ASSESS AGILITY: AN AGILITY ASSESSMENT APPROACH SUPPORTED WITH AN AUTOMATED WEB BASED AGILITY ASSESSMENT TOOL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

ONAT EGE ADALI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

MARCH 2017

ASSESS AGILITY: AN AGILITY ASSESSMENT APPROACH SUPPORTED WITH AN AUTOMATED WEB BASED AGILITY ASSESSMENT TOOL

Submitted by ONAT EGE ADALI in partial fulfillment of the requirements for the degree of **Master of Science in The Department of INFORMATION SYSTEMS Middle East Technical University** by,

Prof. Dr. Deniz ZEYREK BOZŞAHİN _____
Director, **Graduate School of Informatics**

Prof. Dr. Yasemin YARDIMCI ÇETİN _____
Head of Department, **Information Systems**

Prof. Dr. Onur DEMİRÖRS _____
Supervisor, **Information Systems**

Dr. Özden ÖZCAN TOP _____
Co-Advisor, **RSRC & Lero DKIT**

Examining Committee Members:

Assoc. Prof. Dr. Aysu BETİN-CAN _____
Information Systems, Middle East Technical University

Prof. Dr. Onur DEMİRÖRS _____
Information Systems, Middle East Technical University

Prof. Dr. Ali DOĞRU _____
Computer Engineering, Middle East Technical University

Assoc. Prof. Dr. Altan KOÇYİĞİT _____
Information Systems, Middle East Technical University

Asst. Prof. Dr. Ayça TARHAN _____
Computer Engineering, Hacettepe University

Date: 31.03.2017

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Onat Ege, Adalı

Signature : _____

ABSTRACT

ASSESS AGILITY:

AGILITY ASSESSMENT APPROACH SUPPORTED WITH AN AUTOMATED WEB BASED AGILITY ASSESSMENT TOOL

Adalı, Onat Ege

MSc., Department of Information Systems

Supervisor: Prof. Dr. Onur Demirörs

Co-Advisor: Dr. Özden Özcan Top

March 2017, 244 pages

Today, more and more organizations are adopting agile methodologies to their software development processes. However, this adoption process is not straightforward due to the extensive knowledge and effort required. Currently, most agile adopters use assessments at a regular basis to understand the extent of their agility and to determine the success of their agile adoption. There have been many studies conducted about agile assessment models in the literature. Still, these models require significant time and effort to apply. The amounts of time and effort required to perform agility assessment with defined agility assessment models could be decreased with tool usage. However, existing agility assessment tools do not meet expected criteria for organizations to determine their agility. Therefore, there is a need for a generic agility assessment tool that automates and guides the agility assessment process fully and is reachable for all organizations through the web infrastructure. In this thesis, an agility assessment approach supported with an automated web-based agility assessment tool is developed. The study first explores the existing agility assessment tools, then examines the most prominent features of those tools, deducts expected criteria from the features, and evaluates the tools based on the expected criteria. After that the study presents an exemplar agility assessment process that defines the rules to perform and agility assessment with a comprehensive agility assessment reference model. Then, presents the requirements and design of the tool. Afterwards, the details of a multiple case study that conducted to evaluate the tool is presented. Finally, conclusion and future work is given with details.

Keywords: Agility Assessment Tool, AssessAgility, AgilityMod, Automation, Agile Software Development

ÖZ

ASSESSAGILITY:

OTOMATİZE EDİLMİŞ WEB TABANLI BİR ÇEVİKLİK DEĞERLENDİRME ARACI İLE DESTEKLENMİŞ BİR ÇEVİKLİK DEĞERLENDİRME YAKLAŞIMI

Adalı, Onat Ege

Yüksek Lisans, Bilişim Sistemleri Bölümü

Tez Yöneticisi: Prof. Dr. Onur Demirörs

Eş Danışman: Özden Özcan Top

Mart 2017, 244 sayfa

Günümüzde çevik metodolojileri benimseyen organizasyonların sayısı gittikçe artmaktadır. Ancak bu benimseme süreci çokça bilgi ve efor gerektirdiğinden dolayı kolay değildir. Mevcut durumda, çevik metodolojileri benimseyen kurumların çoğu, çevikliklerinin ölçüsünü ve çevik benimsemelerinin başarısını belirlemek için değerlendirmeleri kullanmaktadırlar. Literatürde, çevik değerlendirme modelleri ile ilgili birçok çalışma mevcuttur. Ancak, bu modellerin uygulanması için de yine önemli miktarda zaman ve efor gerekmektedir. Tanımlı çevik değerlendirme modelleri ile çeviklik değerlendirmeleri gerçekleştirirken gereken zamanı ve eforu düşürmek araç kullanımı ile mümkün olabilir. Ancak, var olan çeviklik değerlendirme araçları kurumların çevikliklerini belirlemek için gereken kriterleri sağlamamaktadırlar. Bu sebeple web altyapısı üzerinden tüm kurumlar tarafından ulaşabilen genel bir çeviklik değerlendirme aracına ihtiyaç vardır. Bu tez çalışmasında, otomatize web tabanlı bir çeviklik değerlendirme aracı ile desteklenmiş bir çeviklik değerlendirme yaklaşımı sunulmuştur. Çalışmada öncelikle var olan çeviklik değerlendirme araçları ortaya çıkartılıp, bu araçların üstün özellikleri inceledikten sonra, bu özelliklerin üzerinden beklenen kriterler ortaya konulmuştur ve bu kriterlerle araçları değerlendirilmiştir. Ondan sonra kapsamlı bir çeviklik değerlendirme modeli ile değerlendirme yapmak için gereken kuralları tanımlayan örnek bir çeviklik değerlendirme süreci ortaya koyulmuştur. Sonra, aracın gereksinimleri ve tasarımı tanımlanmış. Daha sonra da geliştirilen aracı değerlendirmek üzere tasarlanan bir çoklu durum çalışmasına yer verilmiştir. Son olarak da sonuç ve gelecekteki çalışmaların detaylarına yer verilmiştir.

Anahtar Sözcükler: Çeviklik Değerlendirme Aracı, AssessAgility, AgilityMod, Otomasyon, Çevik Yazılım Geliştirme

Dedicated to my beloved father ***Abdülrezzak Adalı***

ACKNOWLEDGMENTS

First and foremost, I want to share my deepest gratitude for my advisor Prof. Dr. Onur Demirörs. I have always felt privileged to be able to work with him and I want to thank him for the great mentor he has been to me. His positive attitude, brightest ideas, great support, and unparalleled encouragement widened my vision and enabled me to take firm steps to where I am today. His teachings will always stay with me and he sure will continue to inspire me.

Secondly, I would like to thank to my co-advisor Dr. Özden Özcan-Top for making this study possible. I have always admired her impeccable work and I appreciate the support, guidance, and help she has provided me throughout my studies. However, the most important thing she provided me during this process was her friendship which I am hoping to have for the years to come. It was a pleasure to be able to work with Özden.

I would also like to thank to Dr. Alpay Karagöz for setting me off to the right track, without him I would not be able to accomplish all this. Also, I would like to thank Ebru Gökalp, Ali Mert Ertuğrul, and Ozan Raşit Yürüm for their friendship throughout my master's studies. I would also like to thank to my friends Ayşe Begüm Kılıç and Başak Oğuz for making our miserable and cold institute more bearable.

I would like to thank my best friends Ramazan Kürkan and Çınar Turgut for all the unimaginable ways they have helped me throughout this study. They might not agree with me but I will always insist to say that they are the best team mates and friends anyone can have. I also want to thank my other best friends Hakan Altındiş, Çağlar Aydoğan, and Burak Atmaca for sharing amazing adventures with me. Knowing you all is a great pleasure.

I would like to thank to my old friend Semih Kanalıcı for always being there for me. His understanding and never-ending encouragement were beyond imagination.

Finally, I would like to thank to my family. I cannot put into words how thankful I am for my father therefore, I would simply thank my father Abdülrezzak, for just being my dad because he is simply the best. I want to thank to my mother Sibel for giving me the things I thought I lost. Also, I want to thank to my two brothers Oğulcan and Ata for being the joys of my life. I really appreciate my extraordinary family's love and compassion.

I don’t believe that when you lost someone, their presence isn’t felt. Hence, my final thank is for my mother. I would like to thank her for making me who I am and I know that she was and will continue to be with me.

This research is supported by Scientific and Technological Research Council of Turkey (TÜBİTAK), grant number 113E528.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

A	Available
ASP	Agile Software Process Model
CRUDL	Create, Read, Update, Delete and List
DevOps	A Clipped Compound of Development and Operations
DSDM	Dynamic Systems Development Method
ER	Entity Relationship
FDD	Feature-Driven Development
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
iOS	iPhone Operating System
IOT	Internet of Things
IP	Internet Protocol
ISO	International Organization for Standardization
MSS	Main Success Scenario
N/A	Not Available
RQ	Research Question
RUP	Rational Unified Process
SAAS	Software as a Service
SME	Small and Medium Enterprises
SNA	Social Network Analysis
SPICE	Software Process Improvement and Capability dEtermination
SQL	Simple Query Language
SRS	Software Requirements Specification
Std	Standard
TCP	Transmission Control Protocol
UCD	Use Case Diagram
XP	eXtreme Programming

CHAPTER 1

INTRODUCTION

This chapter presents the introductory statement about this study. First, the background of the problem is given in details, to set up the grounds for the proposition. Consequently, the problem statement is provided to clarify the proposition in details. After that, the significance of the study is given to describe the reasons for conducting the study. Then, the objectives of the study are given to describe the generic and specific goals of the study. Accordingly, the research strategy section is given to describe the methodology to achieve the goals. Finally, the structure of the thesis is provided to describe the organization of the thesis.

1.1. Background of the Problem

Agile Methodologies have emerged to overcome the inherent challenges of the traditional software development methodologies [1, 2]. These challenges can be summarized as: upfront planning, strong reliance on processes and tools, heavy documentation, command and control culture, conformation to plan and predictive approach to development [3]. In contrast to the traditional methodologies, Agile methodologies are low on ceremony, more people-centric, less formal, iterative, and collaborative-based [4]. Nerur et al. described the differentiating points between traditional and agile methodologies as: approach to control, management style, knowledge management, role of the customer in development process, role assignment, communication style, development life-cycle, organizational culture and technology [5]. To address these points, Agile Methodologies offer a set of principles and practices based on the four Agile values and twelve supporting principles included in the Agile Manifesto [6]. From 1990s till today, many Agile Software development methodologies have been developed: Scrum [7], Dynamic Systems Development Method (DSDM) [8], Agile Software Process Model (ASP) [9], Crystal Family [10], eXtreme Programming (XP) [11], Adaptive Software Development (ASD) [12], Feature-Driven Development (FDD) [13] and Lean Software Development [14]. These methodologies are designed with customization in mind; they offer the flexibility to adopt a subset of principles and practices, based on the culture, values, and needs of an organization or a team [15].

Today, many organizations are transitioning from traditional methodologies to Agile. However, the transition process is not straightforward [16, 17] as it might seem. In most cases, transitions end up with adoption of random principles and practices. Hence, the benefits afforded by the agile practices are not fully realized [18] and adopters struggle to become more Agile; i.e. increase their Agility. For this reason, adopters need to be aware about the situation of their transition and make sure that they are heading in the right direction. The best way to attain this, is to continuously evaluate the software processes and apply improvement actions consequently. The first form of a documented agility assessment was given by Boehm and Turner [1] and since then many researchers and practitioners have developed different forms of agility assessment approaches. These approaches include; agility assessment models such as *Agile Maturity Model* [19], *Agile Adoption Framework* [20], *Agile Scaling Model* [21], *Benefield's Model* [22] and *Scrum Maturity Model* [23], agility checklists such as *Scrum Checklist* [24], *Joe's Unofficial Scrum Checklist* [25], and *Corporate Agile 10-Point Checklist* [26] , and agility surveys such as *Agile Enterprise Survey* [27] and *Agile Maturity Self-Assessment Survey* [28].

The variety of the agility assessment approaches sounds promising but their application is impeded by the significant amount of time and effort spent to comprehend and perform them. This situation led the introduction of agility assessment tools that have the ability to guide and automate the agility assessment process. Independent from their forms, assessment tools are designed with the objectives of helping an assessor perform an assessment in a consistent and reliable manner, reducing subjectivity and contributing to the achievement of valid, useful and comparable assessment results and performing the assessment more efficiently [29]. For the time being, many agility assessment tools have been proposed with increasing the applicability and efficiency of the agility assessments in mind, however, their capability to meet the objectives for an assessment tool is still a question of debate.

1.2. Statement of the Problem

Assessing agility is crucial for agile adopters to understand the current status of their transition and discern the gaps towards agility. However, grasping and applying agility assessment approaches such as assessment models, checklists, and surveys requires significant amount of time and effort. As a consequence, tool support for agility assessments are at vital importance due to their ability to increase the efficiency of the assessments.

However, majority of the agility assessment approaches do not provide automation facilities for adopters to take advantage from. Moreover, the provided ones are mostly focused on conducting the assessment but lacking the support for other important parts of the assessment process such, as planning and data validation. Therefore, to increase the efficiency of the agility assessment process, tools are

required to have the features to facilitate and automate the whole assessment process including planning, conducting, and reporting the agility assessments.

Furthermore, current agility assessment tools are narrowly-scoped in terms of covering Agile principles and values due to the set of agile practices that they use to indicate the level of agility. While these practices are crucial for specific implementations of agile methods, the mere absence or presence of these practices is not sufficient to indicate the level of agility. Also, majority of the approaches only work with specific methodologies such as Scrum or XP, and implementations such as Kanban or DevOps. As a consequence, they fail to provide an indication of agility levels or the possible improvement areas towards agility. Therefore, there is a need for a fully automated agility assessment tool, that provides a comprehensive way of assessing Agility without relying on a specific Agile Methodology, an implementation, or a limited set of practices.

1.3. Significance of the Study

Agility assessments are crucial instruments to assess the status of an adopted Agile Methodology. These assessments enable adopters to see if their methodology has met its intended purpose and to determine the improvement areas for increased agility. However, performing these assessments without tool support is time and effort consuming. Many tools have been developed to address this problem, although none of them has been able to guide and automate the whole assessment process. The tool proposed in this study will be able to provide guidance and automation through an agility assessment's planning, conducting, and reporting stages. Additionally, the tool will be based on a structured and method-independent agility assessment model that provides clearly defined agility levels and comprehensive set of practices compatible with the Agile Manifesto. This will enable the tool to yield clear and understandable results and also enable the tool to be generic which means that it has the ability to be used upon any method, context, or implementation.

The guidance capability of the proposed tool will aid the assessors before, during, and after the agility assessment process and the automation capability will aid organizations to perform assessments in an inexpensively, timely and effortlessly manner.

Apart from the contributions proposed with the tool, the study presents an exemplar agility assessment process that describes how to conduct assessments with the agility assessment reference model that the tool is based on. The exemplar process will extend the model and it will also improve the applicability of it by providing guidance for both researchers and practitioners.

Moreover, the study will provide an extensive list and comparison of the currently available agility assessment tools for the researchers and practitioners. Consequently, enabling them to learn the limitations and characteristics of the current tools and

choose a subsequent tool for their own needs or studies. Subsequently, the study will yield benefits of using a tool for agility assessments.

For future researchers, this study will provide the baseline for the expected criteria for an agility assessment tool to help them develop and improve their own assessment tools.

1.4. Objectives of the Study

The main focus of this study is to develop a web-based online agility assessment tool that has the ability to guide and automate the agility assessment process; including planning, conducting, and reporting phases of the agility assessment process.

Accordingly, the first objective of the study is to review and evaluate existing agility assessment tools to determine their suitability for agility assessments and to bring out the expected criteria for an agility assessment tool should have by examining their features.

The second objective of the study is to define the requirements for the purposed agility assessment tool and develop a design for it, based on the requirements determined. And then, build the agility assessment tool based on the requirements gathered and design created.

Another objective of the study is to define an exemplar agility assessment process that will guide the practitioners to conduct assessments with the tool.

The final objective of the study is to evaluate the proposed tool's and exemplar agility assessment process' suitability for agility assessments so that they could be actively used in the industry and future studies can be conducted to improve the tool support for agility assessments based on the findings.

1.5. Research Strategy

As a whole, the research strategy followed through this study can be categorized as constructive research [30] since it involves "building an innovation based on the existing knowledge and new technical or organizational advancements" as Järvinen suggests. However, at the beginning of the study we did not have the existing knowledge concerning the agility assessment tools due to the low number of research about the subject. Therefore, to build up the knowledge, the activities shown in the upper side of the dashed line in the Figure 1 are performed. First, we have conducted a literature review to find out the existing agility assessment tools. Next, we have conducted a multiple case study to understand their capabilities and characteristics. After that we have defined the requirements for an agility assessment tool based on the findings of the literature review. Then, we have used our existing research and knowledge on the AgilityMod: Agility Assessment Reference Model to build an

exemplar assessment process that will serve as the basic workflow of the tool that will be developed.

Remaining activities that falls below the dashed line on Figure 1, can be categorized as design science [31] as suggested by the research framework of March and Smith. According to their classification, there are two types of science: the first one, the natural science, tries to understand the reality and the second one, the design science, "attempts to create things that serve human purposes". The basic activities of design science are summarized as "build" and "evaluate". Järvinen [30] describes the activities in the build phase as "a process of constructing an artefact/innovation for a specific purpose" and the activities in evaluate phase as "a process of determining how well the artefact performs". In the light of that, first we have built the AssessAgility with the purpose of developing an agility assessment tool and then evaluated it with a multiple case study to see how it performs. Finally, with the results from the multiple case study we have updated our tool.

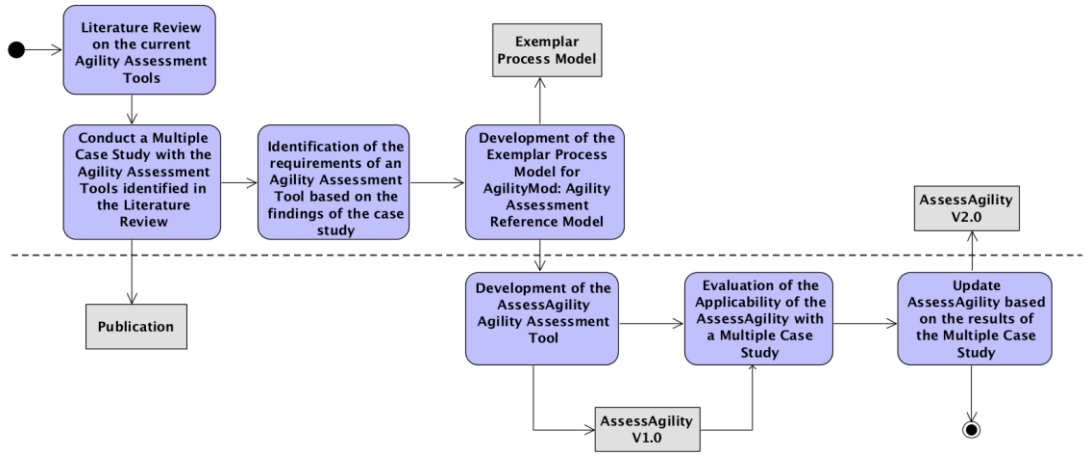


Figure 1: Research Strategy

1.6. Structure of the Thesis

Chapter two includes the review about the related literature on agility assessment tools. The chapter provides an extensive list of the tools and presents detailed descriptions of the agility assessment tools that are templates or software programs. Also, chapter two includes a multiple case study that is conducted to compare and determine the expected features of the agility assessment tools that are presented in details.

Chapter three presents the software requirements specification of the proposed tool including use case scenarios that captures the requirements.

Chapter four presents an exemplar agility assessment process for AgilityMOD reference model which is also utilized during the application of the tool.

Chapter five presents a multiple case study conducted to validate and examine the capabilities of the proposed tool. It includes detailed descriptions of each case and procedures of data collection and analysis. Finally, it gives the detailed results and findings of each case presented in the multiple case study.

Chapter six concludes the thesis with discussion and future work.

CHAPTER 2

LITERATURE REVIEW

This chapter presents the literature review about agility assessment tools that are developed for academic and/or commercial purposes. In the agility assessment tools section, the tools that are found in the literature review are given with their detailed descriptions. In the evaluation and comparison of the agility assessment tools section a multiple case study designed to evaluate and compare the tools is introduced, then the results of the multiple case study are presented with details.

2.1 Agility Assessment Tools

The agility assessment tools can be described as templates or software programs that are developed to automate and guide the agility assessment process. The main intent of these tools is to measure to what extent a software company has succeeded to adapt to agile methods in its software development process [32]. The tools are developed based on agile assessment models and methods which are designed to measure the agility level of an agile method adoption.

In the literature review agility assessment tools that are developed for academic and/or commercial purposes are searched upon world wide web and scientific databases such as IEEE Xplore, SpringerLink and Web of Science. In the searching process, we reviewed the main concepts and terminology in the agility assessment domain and identified the primary keywords. Then we checked synonyms, alternative usages and hypernyms for each keyword. Finally, we formulated a search string by using Boolean operators (‘AND’ and ‘OR’) with wildcard character (‘*’). The search query used in the searching process is given below:

“(((Agil*) AND (Assess* OR Evaluat*) AND (Tool OR Survey OR Model OR Checklist OR Approach OR Method))”

The list of the tools attained as the result of the searching process are given at the following table.

Table 1: List of Agility Assessment Tools

	Name	Owner	Type	Availability
1	A Corporate Agile 10-point Checklist [26]	Elena Yatzeck	Text Based	A
2	abetterteam.org [33]	Sebastian Hermida	Web Tool	N/A
3	Agile 3R Model of Maturity Assessment [34]	Phani Thimmapuram	Text Based	A
4	**Agile Assessment [35]**	**Piotr Nowinski**	**Sheet Tool**	**A**
5	Agile Assessment [36]	Toughtworks	Web Tool	N/A
6	**Agile Enterprise Survey [27]**	**Storm-Consulting**	**Web Tool**	**A**
7	Agile Essentials[37]	Ivar Jacobson International	Card Game	A
8	Agile Health Assessment Tool [38]	Agile Transformation Inc.	Chart Tool	Paid Service
9	**Agile Health Dashboard [39]**	**Len Lagestee**	**Sheet Tool**	**A**
10	**Agile Journey Index [40]**	**Bill Krebs**	**Sheet Tool**	**A**
11	Agile Maturity Matrix in JIRA [41]	Atlassian	Web Tool	Paid Service
12	Agile Maturity Self Assessment [42]	Robbie Mac Iver	Text Based	A
13	Agile Maturity Self-Assessment Survey [28]	Eduardo Ribeiro	Text Based	A
14	**Agile Process Assessment Tool [43]**	**Info Tech Research Group**	**Sheet Tool**	**A**
15	Agile Readiness Questionnaire [44]	PM Gadget	Text Based	N/A
16	**Agile Self Assessment [45]**	**Cape Project**	**Web Tool**	**A**

	Name	Owner	Type	Availability
		Management		
17	Agile Self Assessment [46]	Mike Hoogveld	Web Tool	N/A
18	Agile Team Evaluation [47]	Eric Gunnerson	Text Based	A
19	**Agility Questionnaire [48]**	**Marcel Britsch**	**Sheet Tool**	**A**
20	AGIS [49]	Santiago Matalonga	Text Based	A
21	Borland Agile Assessment [50]	Borland	Text Based	A
22	CAMT (Comprehensive Agility Measurement Tool)[51]	Ameya S. Erande, Alok K. Verma	Unknown	N/A
23	Checklist for Change Agents [52]	Michael Sahota	Text Based	A
24	**Comparative Agility Tool [53]**	**Mike Cohn and Kenny Rubin**	**Web Tool**	**A**
25	**Depth of Kanban [54]**	**Christophe Achouiantz**	**Chart Tool**	**A**
26	Enterprise Agile Practice Assessment Tool [55]	DrAgile	N/A	Paid Service
27	**Enterprise Agility Maturity Matrix [56]**	**Eliassen Group**	**Sheet Tool**	**A**
28	Forrester's Agile Testing Maturity Assessment Tool [57]	Diego Lo Giudice, Margo Visitacion, Phil Murphy, Rowan Curran	N/A	Paid Service
29	**GSPA: A GENERIC SOFTWARE PROCESS ASSESSMENT TOOL [58]**	**Ozan Raşit Yürüm**	**Desktop Tool**	**A**

	Name	Owner	Type	Availability
30	How Agile Are You? [59]	Kelly Waters	Text Based	A
31	**IBM DevOps Practices Self Assessment [60]**	**IBM**	**Web Tool**	**A**
32	Joe's Unofficial Scrum Checklist [25]	Joe Little	Text Based	A
33	Lean Enterprise Self-Assessment Tool [61]	Lean Advancement Initiative (LAI)	N/A	N/A
34	Maturity Assessment Model for Scrum Teams [62]	Marmamula Prashanth Kumar	Text Based	A
35	Net Objectives Lean-Agile Roadmap for Achieving Enterprise Agility [63]	NetObjectives	N/A	N/A
36	**Open Assessments [64]**	**Scrum Inc.**	**Web Tool**	**A**
37	Readiness & Fit Analysis [65]	Suzanne Miller	Text Based	A
38	ReadyForAgile Part 1 and Part 2 [66]	Salah Elleithy	Web Tool	N/A
39	Scrum Assessment Series [67]	David Hawks	Text Based	A
40	Scrum Butt Test (Nokia Test) [68]	Bas Vodde, Jeff Sutherland	Text Based	A
41	Scrum Checklist 2012 [69]	Boris Gloger	Text Based	A
42	SPIALS: A light-weight Software Process Improvement Self-Assessment Tool [70]	Disorn Homchuenchom, Chayakorn Piyabunditkul, Horst Lichter, Toni Anwar	Desktop Tool	N/A
43	Squad Health Check Model	Henrik Kniberg	Text Based	A

	Name	Owner	Type	Availability
	[71]			
44	Team Barometer (Self-Evaluation Tool) [72]	Jimmy Janlén	Card Game	A
45	**TeamMetrics [73]**	**Christiaan Verwijs**	**Web Tool**	**A**
46	The Unofficial Scrum Checklist [74]	Henrik Kniberg	Text Based	A

The list above includes 20 tools that are text based checklists, questionnaires, and card games which are nor templates nor software programs that have the ability to automate and guide the assessment process. Furthermore, 12 of the remaining 26 tools are out of reach due to their unavailability. Therefore, it is decided that giving the detailed descriptions of the remaining 14 tools is relevant for this work. The detailed descriptions of the tools are given below in alphabetical order.

2.1.1 Agile Assessment

The Agile Assessment aims to assess the agility level of a team or organization and uncover improvement areas. The assessment is grounded on another assessment presented by Dean Leffingwell in Scaling Software Agility: Best Practices for Large Enterprises [75] and it consists of 66 statements/questions grouped in 7 areas: product ownership, agile process, team, quality, engineering practices, fun & learning and integration [35]. Each statement/question is answered on a 5-point scale and tool generates a percentage score for each of the 7 areas. The tool is available in spreadsheet format and it's free to use.

2.1.2 Agile Enterprise Survey

Agile Enterprise Survey is a web-based online survey designed by Storm Consulting to assess enterprise agility. The survey presents different sets of statements and asks the assessor to specify how well these statements reflect his or her organization. The statements are placed under 16 questions that are categorized into five distinct parts namely: Values and Practices, Working Environment, Capabilities, Activities, Blue Sky thinking, and Organization Background. The survey can be run externally and is anonymous. No names or email addresses are stored with the survey data.

2.1.3 Agile Health Dashboard

Agile Health Dashboard is a tool that is designed to measure the team agility. The tool consists of following parts: Completed/Committed Stories, Team Composition, Team Size, Team Member Dedication, and Family Fun. These parts are updated after every sprint to observe the current team health. The tool comes in a spreadsheet format with pre-defined features and metrics to reflect the team agility.

2.1.4 Agile Journey Index

Agile Journey Index is an agility assessment model that aids organizations to improve their agility. The index is constructed around 19 questions in 4 distinct groups. “Questions are related to 19 key agile practices and groups include the following: Planning, Do, Wrap, and Program Organization Criteria”[76]. “Each practice is rated on a scale of 1-10, with specific criteria for each number” [76]. The tool is available for use in a spreadsheet format and provided with supplementary documentation.

2.1.5 Agile Process Assessment Tool

The Agile Process Assessment Tool which was developed by Info-Tech Research Group analyzes “how well an organization is lined up with the agile ideal across different process issues” [43]. The tool includes 67 questions in six different categories. These categories are: Configuration Management, Change Management, Release Management, Comprehensive Testing, Automation, and Compliance. The tool is available in spreadsheet format and it includes instructions encapsulated in the spreadsheet as well.

2.1.6 Agile Self Assessment

Agile Self Assessment tool developed by Cape Project Management, Inc. is a web based online survey that is built upon the Scrum Checklist [74]. The tool reflects the results of the checklist onto an agility maturity matrix [56] that has five different levels of agility. There are 60, agree and disagree questions. “The scoring of the questions are based upon the overall importance of the answer” [56]. After answering the questions, the tool calculates the scores and indicates the agility level according to the agility maturity matrix.

2.1.7 Agility Questionnaire

Agility Questionnaire allows “establishing a holistic view of organizational, team and project related factors, thus creating an Agility Profile which provides the necessary insight to make the right decision towards delivery methodology and more importantly areas of the methodology that require tailoring to optimize for the specific case at hand”. “The questionnaire consists of two parts: Agility, and Project Profile. The questions under Agility are used to assess the capability to be Agile and the questions under Project Profile, indicates the characteristics of a particular project that may be used for tailoring methodologies” [48]. The tool comes in a spreadsheet format and enables answering questions and identifying agile capability.

2.1.8 Comparative Agility

The main idea behind Comparative Agility [77] is assessing an organizations’ agility by comparing it to its competitors. “Rather than guiding organizations to a perfect score or level of agility, it presents a comparison of results” [78]. The accompanying agility assessment tool is a web based online survey that is designed for self-assessments. The assessment includes 100 questions that are divided into seven dimensions. The dimensions are: teamwork, requirements, planning, technical practices, quality, culture, and knowledge creation [78]. Each dimension includes three to six characteristics and each characteristic has distinct questions related to it.

2.1.9 Depth of Kanban

Depth of Kanban, is a coaching tool for assessing the depth of a Kanban [79] implementation. The tool is a spider graph that is structured around seven axes that are based on Kanban principles. The principles are: Visualize, Limit Work in Progress, Manage Flow, Make Policies Explicit, Implement Feedback Loops, Improve, and Effects. Each axe includes different numbers of yes/no questions to answer and the depth of the implementation (the level of agility) is determined by the positive answers given.

2.1.10 Enterprise Agility Maturity Matrix

The primary goal of Enterprise Agility Model is to "encapsulate and document the well-known best practices for transforming an Enterprise to Agile as simply as possible, inventing as little new as possible" [80]. The model is structured according to the principles of Agile Manifesto and the Enterprise Agility Maturity Matrix tool is provided with the model. "The tool is mainly used for setting transformation goals, monitoring progress, and getting everybody on the same page regarding Agile including: Agile Coaches, team members, managers, and senior leadership"[56]. The tool is in the spreadsheet format and comes in a compressed file including supplementary documentation about both Enterprise Agility Model and the Enterprise Agility Maturity Matrix tool. The matrix is also integrated to the JIRA Software [81] project management tool but since it's a paid-service, it's excluded from the scope of this work.

2.1.11 GSPA: A Generic Software Process Asssessment Tool

GSPA is an offline process assessment tool that enables making process assessments with a wide range of process assessment models including the agility assessment models. The tool employs a meta-model that combines common structures of most common process assessment models: CMMI [82] and ISO 15504 [83]. By using the meta-model, any kinds of assessment models can be introduced into the tool. Since the tool does not include a predefined agility assessment model or survey, we have used tool's meta-model to integrate one of the structured and complete agility assessment models, AgilityMOD [84, 85] that guides organizations on their way to become agile into the tool and per- formed our evaluation according to it.

2.1.12 IBM DevOps Practices Self-Assessment

The IBM DevOps Practices Self-Assessment is developed to "evaluate the state of an organization's software delivery approach" [60]. The aim is to improve agility with adoption paths and proven practices. Web based online tool enables assessors choose an adoption path and assess the organization according to the practices related to the chosen path.

2.1.13 Open Assessments

Open Assessments is a resource that Scrum.org [86] offers freely for software development community. Open Assessments consists of 4 different assessments called; Scrum OPEN, Nexus OPEN, Product Owner OPEN and Developer OPEN. The assessments allows practitioners to gauge their basic knowledge of Scrum, the structure of a Scrum Team, and the fundamentals of scaling Scrum [64]. The

assessments are designed as initials blocks to prepare assessment takers to professional level assessments and certificates. Scrum OPEN aims to assess basic scrum knowledge and it consists of 30 questions which can be answered within 30 minutes of time limit. Developer OPEN aims to assess the knowledge of basic development practices used on a Scrum Team and again it consists of 30 questions which can be answered within 30 minutes of time limit. Nexus OPEN aims to assess assessment takers' understanding of the Nexus Framework and it consists of 15 questions which can be answered within 25 minutes of time limit. Finally, Product Owner OPEN aims to assess basic knowledge of the Product Owner role in Scrum and it consists of 15 questions which can be answered within 30 minutes of time limit.

2.1.14 TeamMetrics

TeamMetrics is a simple, free survey tool to help (Agile) teams improve by gathering objective data about key team factors such as team morale and interpret the results with the help of benchmarks [73]. When a survey is created, a link is also generated for participants to join the survey. After joining the survey, each participant answers dichotomous questions and submit their results. Then, the tool generates a report based on the answers and displays the team's situation according to the chosen set of metrics. The tool uses 2 different set of metrics namely; Motivational Potential and Team Morale. The Motivational Potential set aims to measure a job's motivating potential and it measures five dimensions; skill variety, task significance, task identity, autonomy and feedback. The set is developed on the Job Diagnostic Survey (JDS) [87]. The Team Morale set aims to measure the morale of the team and it is based on Utrecht Work Engagement Scale (UWES) [88].

2.2 Evaluation and Comparison of the Agility Assessment Tools

For the evaluation and comparison of the agility assessment tools we have designed and conducted a multiple case study which consists of assessments made with each of the 14 tools described in the previous section. The details of the multiple case study are given in the section 2.2.1 Multiple Case Study and the results of the study is given at section 2.2.2 Case Study Results.

2.2.1 Multiple Case Study

The main objective of the multiple case study is to determine current tools' capability to meet the expected criteria for an agility assessment tool. However, due to lack of study on this subject the expected criteria are unknown. Therefore, another objective of the multiple case study is to review the existing agility assessment tools to determine their features and deduct expected criteria for an agility assessment tool expected to have. In order to achieve these purposes, we defined the following research questions (RQ):

RQ1: What features do current agility assessment tools have and what are the expected criteria deducted from those features?

RQ2: To what extent are the current agility assessment tools sufficient to meet the expected criteria?

Case Study Design

To answer the research questions, a software development organization that employs an established agile software development methodology, namely Scrum [7], was selected. The selected organization for the case study is currently developing mobile and web applications with two dedicated self-organizing teams, each consisting of 5 team members. For the project level assessments, the selected project for the study was a web application, which was completed within six weeks-time with a fixed budget. For the team level assessments, the same team that developed the project was selected. The selected team consisted of 1 team leader, 3 software developers and a user experience designer. The team has an experience of employing agile software development practices for 4 years.

Within the scope of the case study, 14 agility assessments; for each one of the 14 agility assessment tools were planned. Each assessment was planned according to the target scope established by the accompanying tool. Therefore, assessments were subject to different extends of the organization such as project, team and/or whole organization.

Conduct of the Case Study

The assessments within the study were conducted by the Scrum Master of the selected team who is also the author of this thesis. First, papers, articles, instruction manuals and other documentation about each tool was reviewed before and during each assessment. Then the features of each accompanying tool were listed and recorded with the agility level results. Finally, after the assessments were completed, the results were reviewed with other team members to validate resulting agility levels.

2.2.2 Findings of the Case Study

2.2.2.1 Findings Related to Research Questions

RQ1: What features do current agility assessment tools have and what are the expected criteria deducted from those features?

To answer the RQ1, the features extracted from related documents were combined with the features captured during the execution of the tools. The features gathered from the agility assessment tools are listed below:

- **Guidance Capability:** Ability to provide guidance for assessors who are not experts on agile software development.

- **Assessment Recording:** Ability to record agility assessment findings and the resulting reports for further modifications, analysis, and comparison.

- **Automated Reporting:** Ability to automatically generate reports for the presentation of the results of the performed assessment.

- **Results Comparison:** Ability to enable comparison between the reports of previously performed assessments.

- **Different Modes of Usage:** Ability to support different usage mods for individuals, multiple users, and parallel assessments.

- **Different Scopes Support:** Ability to perform assessments on project, team, and/or organizational levels.

- **Extensibility:** Ability to provide extensibility to meet emerging needs of different types of assessment contexts.

In the light of these features and from the opinions of the experts in agility assessment, the expected criteria for an agility assessment tool were determined as follows:

Coverage:

Agility assessment tools should address all twelve agile principles stated in the Agile Manifesto [89], in order to perform a comprehensive and complete agility assessment. Agile principles and values construct a foundation of agile sense together and explain how agile practices work in practice [90], therefore full coverage of these values and principles are mandatory for agile assessment tools. We rated this criterion based on tools' coverage of 12 agile principles using a four-point ordinal (N-P-L-F) scale. The details of the rating are given below:

- Not Achieved: 0-2 principles are covered
- Partially Achieved: 2-6 principles are covered
- Largely Achieved: 6-11 principles are covered
- Fully Achieved: 12 principles are covered

Availability:

In order to provide equal access and equal opportunity for majority, an agility assessment tool should be universally reachable. Therefore, tools are expected to be online and web-based applications. This criterion is evaluated with a dichotomous (Web-Based/Not Web-Based) scale.

Guidance Capability:

Agility assessment tools are expected to provide guidance for assessors who are not experts on agile software development. In this manner, tools should include guiding facilities such as help menus, example cases and responses, tips, and samples to guide assessors both beforehand and during the assessment. This criterion is

evaluated with a four-point rating (N-P-L-F) scale according to the three categories of guidance capability expected from the tools. The categories are: providing guidance before assessment, providing guidance during the assessment, and providing guidance after the assessment. The details of the rating are given below:

- Not Achieved: None of the guidance capabilities is provided
- Partially Achieved: Only one type of guidance capability is provided
- Largely Achieved: Two types of the guidance capabilities are provided together
- Fully Achieved: All Three types of the guidance capabilities are provided together

Assessment Recording:

Agility assessment tools are expected to provide recording capabilities to store both agility assessment findings and the resulting reports for further modifications, analysis, and comparison. This criterion is evaluated with a dichotomous (yes/no) scale.

Automated Reporting:

Agility assessment tools are expected to include an automated reporting function that generates reports for the presentation of the results of the performed assessment. Assessment findings, which are supported by graphics and tables would be valuable for the interpretation of the results. This criterion is evaluated with a dichotomous (yes/no) scale.

Comparability:

Agility assessment tools are expected to enable comparison between the reports of previously performed assessments. Continuous learning is a significant part of agile philosophy. It is obvious that agile teams would benefit comparison of their progress which are held within retrospective meetings mostly. An agility assessment tool needs to allow comparison of previous appraisal within the team itself. Here is to mention that assessment results would be valuable for the team itself indicating the challenged points. Therefore, parameters like velocity shouldn't be compared between agile teams. This criterion is evaluated with a dichotomous (yes/no) scale.

Different Modes of Usage:

Agility assessments can be performed by single individuals and/or multiple individuals in teams, in departments, or in groups. Hence, tools are expected to support different usage mods for individuals and multiple users, and provide parallel assessments for simultaneous assessments. This criterion is evaluated with a four-point rating (N-P-L-F) scale based upon three types of usage categories. These types are single user assessment mode, multi-user assessment mode and parallel assessment mode. The details of the rating are given below:

- Not Achieved: None of the usage modes is provided

- Partially Achieved: Only one type of usage mode is provided
- Largely Achieved: Two types of the usage modes are provided together
- Fully Achieved: All Three types of the usage modes are provided together

Different Scopes:

An agility assessment may be performed on from different perspectives. Assessments may target projects, teams, and/or organizations. Therefore, Agility assessment tools are expected to be able to support different types of scopes to provide different types of agility assessments. This criterion is evaluated with a four-point rating (N-P-L-F) scale according to the three types of scopes: project, team, and organization. The details of the rating are given below:

- Not Achieved: None of the scope types is supported
- Partially Achieved: Only one type of scope is supported
- Largely Achieved: Two types of the scopes are supported together
- Fully Achieved: All Three types of the scopes are supported together

Extensibility:

Performing agility assessments on different contexts may require adaptation and extension of the agility assessment models. Therefore, tools are expected to provide a means of extensibility on model features to meet emerging needs of different types of contexts. This criterion is evaluated with a dichotomous (yes/no) scale.

RQ2: To what extent are the current agility assessment tools sufficient to meet the expected criteria?

To be able to address the RQ2, 14 assessments were performed with accompanying tools and then each tool was evaluated according to the nine criteria that have obtained in RQ1. The case study, revealed that none of the current agility assessment tools was able to meet all of the expected criteria. The tool that is able to meet the most of the criteria, was the Comparative Agility, with completely satisfying seven out of nine criteria. For the overall performance of each tool, the table on the next page is provided to summarize each tool's sufficiency to meet the criteria defined. Then the detailed findings on each criterion is given below in subsequent subsections.

Table 2: Comparison of the Agility Assessment Tools

#	Tool/Quality Criteria	Coverage	Availability	Guidance Capability	Assessment Recording	Automated Reporting	Comparability	Different Modes of Usage	Different Scopes	Extensibility
1	Agile Assessment	**FA**	Not Web-Based	PA	**Yes**	**Yes**	No	NA	NA	No
2	Agile Enterprise Survey	LA	**Web-Based**	PA	No	No	No	NA	NA	No
3	Agile Health Dashboard	LA	Not Web-Based	**FA**	**Yes**	**Yes**	**Yes**	NA	NA	**Yes**
4	Agile Journey Index	**FA**	Not Web-Based	**FA**	**Yes**	**Yes**	**Yes**	NA	NA	No
5	Agile Process Assessment Tool	LA	Not Web-Based	PA	**Yes**	**Yes**	No	NA	NA	No
6	Agile Self Assessment	**FA**	**Web-Based**	LA	**Yes**	**Yes**	No	LA	NA	No
7	Agility Questionnaire	LA	Not Web-Based	LA	**Yes**	**Yes**	No	NA	**FA**	**Yes**
8	Comparative Agility	**FA**	**Web-Based**	**FA**	**Yes**	**Yes**	**Yes**	NA	NA	**Yes**
9	Depth of Kanban	LA	Not Web-Based	PA	N/A	No	No	NA	NA	**Yes**
10	Enterprise Agility Maturity Matrix	**FA**	Not Web-Based	LA	**Yes**	**Yes**	No	NA	LA	No
11	GSPA	**FA**	Not Web-Based	LA	**Yes**	**Yes**	No	LA	LA	**Yes**
12	IBM DevOps Practices Self Assessment	LA	**Web-Based**	LA	**Yes**	**Yes**	No	NA	NA	No
13	Open Assessments	**FA**	**Web-Based**	**FA**	No	**Yes**	No	NA	NA	No
14	TeamMetrics	NA	**Web-Based**	**FA**	**Yes**	**Yes**	**Yes**	**FA**	NA	No

Coverage:
Findings of the case study revealed that *Agile Assessment*, *Agile Health Dashboard*, *Agile Journey Index*, *Agile Process Assessment Tool*, *Agility Questionnaire*, *Enterprise Agility Matrix*, and *IBM DevOps Practices Self-Assessment* are developed with questions based upon commonly accepted agile practices and applications of these practices. Although commonly accepted practices are compatible with the agile principles and provide some degree of evaluation, a comprehensive set of practices is essential for full coverage of the context. Due to this reason, only *Agile Assessment Tool* provides full coverage based on its comprehensive set of agile practices. *Comparative Agility*, *Agile Self Assessment, and Open Assessment* tools enable assessment based upon Scrum method and these three provide full coverage among other tools assessed. The *Depth of Kanban* is based upon Kanban method and since it's aimed to assess the depth of a Kanban implementation, it lacks the coverage of some agile principles. The *Agile Enterprise* Survey tool is aimed to assess the organizational agility and it has a higher perspective of agility. This higher perspective gives an abstract coverage of agile principles. TeamMetrics only focuses on the motivational potential of team members and team morale, hence it could not provide coverage on any agile principles. The only tool that provides a structured agile assessment approach is the *GSPA* tool that relies upon AgilityMOD (Agility Assessment Reference Model) [84, 85] it manages to provide a full coverage.

Availability:
The majority of the tools are not web-based tools. Some of these tools are available in spreadsheet format, these are: *Agile Assessment*, *Agile Health Dashboard*, *Agile Journey Index*, *Agile Process Assessment Tool*, *Agility Questionnaire*, *Enterprise Agility Matrix*. *Depth of Kanban* tool is available in printable format. *GSPA* tool is available as an executable JAR file and it is not a web-based application. The web-based tools are *Agile Enterprise Survey*, *Agile Self Assessment*, *Comparative Agility*, *IBM DevOps Practices Self-Assessment, Open Assessment and TeamMetrics*.

Guidance Capability:
All of the tools are able to provide some degree of guidance for assessment process. *Agile Assessment* provides links to the definitions of concepts presented in assessment. *Agile Enterprise Survey* includes clear questions with explanatory notes providing guidance on the top of each question. *Agile Health Dashboard* includes clearly defined data entry fields with explanatory notes attached to them. It also includes example cases and an instruction sheet that includes examples and explanations on how to use the tool. *Agile Journey Index* includes columns that houses guiding notes, examples, explanations, and definitions related to the practices. It also includes sheets that provide example cases. *Agile Process Assessment* tool includes an introduction sheet about the tool and how to use it. *Agile Self-Assessment* survey has an introduction about the structure of the survey and includes a panel that provides navigation to all of the questions. *Agility Questionnaire* includes explanatory columns attached to each question and at the summary section. *Comparative Agility* includes tips and warnings on the top of the questions. It also includes explanatory pop-ups and progress bar that informs the assessor about the state of the assessment. *Depth of Kanban* includes explanatory statements in

questions but it does not have any means of instructions or introduction embedded in the tool. *Enterprise Agility Maturity Matrix* includes clear statements but it has a very limited glossary that houses a single item. *GSPA* includes fields that provide steps about how to use the tool and includes explanatory fields for the models that are used for the assessment. *IBM DevOps Practices Self-Assessment* includes an introductory page and provides warnings and explanations throughout the assessment. *Open Assessments* includes rules and indicators both before and during the assessment and it also provides feedback at the assessment reports. *TeamMetrics* provides instructions and notes to guide both assessment creators and takers before, during and after the assessments.

Assessment Recording:
Except for the *Agile Enterprise Survey* and *Open Assessments*, all the other tools satisfy the assessment recording criterion. This quality criterion is not applicable to *Depth of Kanban* since it's provided in a printable format.

Automated Reporting:
Two of the tools do not provide automated reporting, these are: *Agile Enterprise Survey* and *Depth of Kanban*. All the other tools provide automated reporting functionality with commentary for analysis that is supported with graphical elements such as radar charts, status lights, tables, and bar charts.

Comparability:
Agile Health Dashboard, *Agile Journey Index*, *Comparative Agility* and *TeamMetrics* are the only tools that are able to satisfy the comparability criterion. Agile Health Dashboard enables comparison between different teams, *Agile Journey Index* enables comparison with samples, and *Comparative Agility* provides comparison between a database of surveys. *TeamMetrics* enables comparison between a team member and total score of the team.

Different Modes of Usage:
Only *Agile Self Assessment*, *GSPA,* and *TeamMetrics* provide different modes of usage for multi-users. However, *Agile Self Assessment* and *GSPA* fail to provide parallel assessments for simultaneous assessments and could only provide multi-users by aggregating the results while *TeamMetrics* provides both.

Different Scopes:
Only *Agility Questionnaire* fully provides all three types of different scopes for assessments. Apart from that, *Enterprise Agility Maturity Matrix* and *GSPA* can also provide different scopes for assessments but not completely. *Enterprise Agility Maturity Matrix* provides assessments at organizational and at team levels and *GSPA* provides assessments at project and organization levels.

Extensibility:
In general, web-based tools do not provide any means of extensibility. The only exception is that *Comparative Agility* provides customized surveys by request. Amongst the tools in spreadsheet format only *Agile Health Dashboard* and *Agility Questionnaire* provides explicit extensibility with predefined sections for configuring

and extending the tools. *Depth of Kanban* is extensible in any manner and *GSPA* provides extensibility on process assessment models by its meta-model.

2.2.2.2 Findings Related to the Resulting Agility Levels

In this subsection, the results of the agility assessments that were performed with 14 tools to identify agility level of the software project, team, and/or organization are given. Below, each section describes the assessment scope and approach of the agility assessment made with the resulting agility level indication obtained by using the respective tool. The only tool that we did not manage to get a report was the *Agile Enterprise Survey*. Therefore, the assessment results for *Agile Enterprise Survey* are not included in this section.

Agile Assessment:

Agile Assessment measures agility on seven different dimensions: Product Ownership, Agile Process, Team, Quality, Engineering Practices, Fun & Learning and Integration. The tool designed to get answers from each team member for total of 66 questions. Each question is ranked on a five-point scale while score of 1 indicating the lowest agreement and score of 5 indicating highest agreement. The final results are presented as percentage values and the assessment results for the dimensions are: Product Ownership: 71%, Agile Process: 80%, Team: 87%, Quality: 42%, Engineering Practices: 68%, Fun & Learning: 86% and Integration: 69%.

Agile Health Dashboard:

Agile Health Dashboard indicates agility by assessing a given team's health according to sprint characteristics. After data entry about sprints the dashboard indicated that the assessed team's sprint planning, sprint velocity and team flow health is at the highest level: Excellent.

Agile Journey Index:

Agile Journey Index indicates agility on four different categories: Planning, Do, Wrap, and Program Organization Criteria. However, the tool includes only assessment of first three categories. Each category includes related practices that are rated on a 10-point scale while 1 being the lowest level of agility and 10 being the highest level of agility. The assessment results of the index for the three categories are as follows: Plan: 5.9, Do: 5.0 and Wrap: 3.7.

Agile Process Assessment Tool:

Agile Process Assessment Tool assesses an organization's readiness for agile adoption. It evaluates six different categories that include various statements to rate on a Yes/No or six-point agree/disagree type of scale. The readiness results for agile adoption according to each one of the six categories is shown on a four-point scale Very Low, Low, High, and Very High. The assessment results for the categories are as follows: Configuration Management: Low, Change Management: Low, Release Management: Low, Testing Protocols: Very Low, Automation: Very Low and Compliance: Not Available. The compliance category was not available for the organization that was subject to the case study.

Agile Self Assessment:
Agile Self Assessment tool uses an agile maturity matrix that consists of five levels to indicate agility. The levels are Level 1: Ad Hoc Agile, Level 2: Doing Agile, Level 3: Being Agile, Level 4: Thinking Agile and Level 5: Culturally Agile while Level 1 indicating lowest level of agility and level 5 indicating highest level of agility. After completing 60 questions, the assessment results indicated that the assessed organization is at Level 3: Being Agile.

Agility Questionnaire:
Agility Questionnaire includes two different parts: Agility and Project Profile. The Agility part indicates the assessed organizations agility level and project profile part brings out the characteristics of the project for tailoring agile methodologies. For the case study only the Agility part of the questionnaire is used. The Agility part includes 6 areas that indicate agility on a -10 to 10 scale while -10 being the lowest agility level and 10 being the highest agility level. The results for each area is as follows: Value Focus: 5, Ceremony: 4, Collaboration: -2, Decisions and Information: 2, Responsiveness: 6 and Experience: 4.

Comparative Agility:
Comparative Agility indicates an organization's level of agility in comparison to other organizations that have taken the survey. The results are displayed in a form of standard deviations that shows how given answers differ from the answers given by the competitors. Therefore, positive standard deviations indicate better level of agility and negative standard deviations indicate worse level of agility than competitors. The resulting report includes two graphs: the first one displays the dimension analysis and second one displays characteristic analysis. The tool includes seven dimensions and dimensions are made up of three to six characteristics. Here, only the dimension results are given due to space concerns. The results are as follows: Teamwork: 0.43, Requirements: 0.15, Planning: 0.55, Technical Practices: 0.15, Quality: -0.05, Culture: -0.03, Knowledge Creating: -0.27 and Outcomes: -0.45.

Depth of Kanban:
Depth of Kanban, assesses the agility by identifying the depth of a Kanban implementation. The tool is basically a radar chart that includes seven dimensions. Each dimension includes three different colored areas: red, yellow, light green and dark green. The areas are described from red to dark green as No Improvement, Sustainable Improvement, Excellence and Lean. Each dimension includes different questions and scales. The assessment results for each of the dimensions are in light green: Excellence and the ratings for each dimension is as follows: Visualize: 11, Limit Work in Progress: 3, Manage Flow: 8, Make Policies Explicit: 10, Implement Feedback Loops: 5, Improve: 6, and Effects: 8.

Enterprise Agility Maturity Matrix:
Enterprise Agility Maturity Matrix includes two different levels: organizational level practices and team level practices. The tool includes five-point scale to indicate the levels of agility. The scale is defined as: 0-Impeded, 1-In Transition, 2-Sustainable, 3-Agile and 4-Ideal while Impeded indicating the lowest level and the Ideal

indicating the highest level of agility. The assessment results suggest that on organizational level practices, 10 out of 14 practices are at 3-Agile level and remaining four practices are at 1-In Transition level and on team level 16 out of 35 practices are at 4-Ideal level, 17 out of 35 practices are at 3-Agile level and remaining two practices are at 1-In Transition level.

GSPA (A Generic Software Process Assessment Tool):
GSPA tool has been built upon the AgilityMOD, software agility assessment reference model. AgilityMOD includes two dimensions: Agility and Aspect Dimensions. Agility Dimension includes four levels of agility: Not Implemented, Ad Hoc, Lean and Effective. Aspect Dimension includes four aspects: Exploration, Construction, Transition and Management. AgilityMOD provides guidance for agility assessment of projects and the agility level of a project is determined according to the project teams' ability to perform certain practices defined under each aspect. Teams are given a rating on a four-point rating (N-P-L-F) scale for each aspect. The agility levels of the project based on AgilityMOD are as follows: Exploration Aspect: Effective level, Construction Aspect: Lean level, Transition Aspect: Lean level and Management Aspect: Lean level.

IBM DevOps Practices Self-Assessment:
For IBM DevOps Practices Self-Assessment, an assessment based on the predefined Develop / Test adoption path is performed. The tool employs four levels: Practiced, Consistent, Reliable and Scaled to indicate agility of the each assessed practice. The result of the assessment includes a Blue Border, which indicates a level fully achieved and a Yellow Border, which indicates a level partially achieved. The results of our assessment for each practices assessment in the Develop / Test adoption path are as follows: Design: Blue Border: Reliable & Yellow Border: Scaled, Construct: Blue Border: Practiced & Yellow Border: Consistent, Build: Yellow Border: Practiced, Configuration Management: Yellow Border: Practiced, Assess Quality: Blue Border: Reliable & Yellow Border: Scaled, Test: Blue Border: Practiced & Yellow Border: Consistent.

Open Assessments:
Open Assessments indicates agility by measuring the basic knowledge on Scrum according to the certain roles presented in Scrum development method. In the case study, Scrum OPEN, Development OPEN and Product Owner OPEN assessments are performed. For Scrum OPEN each team member took an individual test and the average of their final results are taken, for Development OPEN each developer took an individual test and the average of their final results are taken and for Product Owner OPEN assessment, only the team member who functions as the Product Owner took the test and his results are taken. For each assessment, a point is given to the correctly answered questions and the results are calculated based on the points earned. For Scrum OPEN average result of all team members is: 26,3/30, for Development OPEN average result of all developers is: 27,6 and for Product Owner OPEN the result of Product Owner is: 13/15.

TeamMetrics:
TeamMetrics, measures key factors: team morale and motivating potential and indicates agility of the team by bringing out team dynamics. For each metric, each team member took anonymous assessments and end results are generated from total sum. The end results are: Team Morale: 8,9/10 and Motivating Potential: 601,5/1000.

Conclusion and Discussion
The results presented in the Table 2: Comparison of the Agility Assessment Tools shows that none of the agility assessment tools that we have evaluated, has the ability to fully comply with the nine expected criteria that we have defined. However, some tools proved themselves useful for special contexts. For example, *Depth of Kanban* is useful for assessing Kanban implementations, *Enterprise Agility Maturity Matrix* is useful for during agile transformations, *Agile Health Dashboard* is useful for monitoring health of agile teams on a sprint basis, and *IBM DevOps Practices Self-Assessment* is useful to adopt a predefined agile adoption path.

In terms of identifying agility, each one of these 14 agility assessment tools has different assessment approaches that yield various different results concerning the agility of the assessed target. Furthermore, tools are mainly developed for assessing agility in certain conditions and contexts such as the beginning of an agile adoption process, certain implementations such as a Kanban implementation, on team, organization and project basis. Therefore, the results of these assessments are inconsistent with each other and comparing the results of these assessments will be irrelevant.

We also observed that majority of the tools use a set of agile practices to indicate the level of agility. While these practices are crucial for specific implementations of agile methods, the mere absence or presence of these practices is not sufficient to indicate the success of the adopted agile method. In addition to that, majority of the tools do not provide an indication of agility levels or the possible improvement areas towards agility. One way to overcome these deficiencies is to build tools that have the capability to support the use of structured agility assessment models that provide clearly defined agility levels and possible improvement areas.

Finally, we observed that tools are mostly focused on conducting the assessment but lacking the support for other important parts of the assessment process such as planning and data validation. Therefore, in addition to fully satisfying our nine criteria and having built upon structured agility assessment models, the tools are expected to have features that facilitate and automate the whole assessment process including planning, conducting, and reporting the agility assessments to reduce the time and effort spent for the assessments.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION AND DESIGN DESCRIPTIONS

This chapter presents the requirements of the tool in accordance with the IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications [91]. The standard is tailored to reflect the most relevant parts of the software of interest. After the requirements, the design descriptions of the tool in accordance with the IEEE Std. 1016-2009, IEEE IEEE Standard for Information Technology— Systems Design— Software Design Descriptions[92]. The standard is tailored to reflect the most relevant design viewpoints and concerns of that viewpoints. including the logical and interface viewpoints are provided with details. The chapter consists of purpose, scope, the product perspective, product functions, specific requirements, logical database requirements and software design descriptions.

3.1 Purpose

Software Requirements Specification (SRS) and Design chapter defines and specifies the software requirements of the tool and provides details about the design decisions made to implement the requirements.

3.2 Scope

AssessAgility is an agility assessment web application which enables its users to determine and understand the agility levels and gaps by performing assessments on their projects. It is based on a structured assessment model: AgilityMOD: Reference Model for Software Agility Assessment, that provides clear indication of where projects stand in terms of agility and which areas of the adopted agile methods need improvement.

Furthermore, since AgilityMOD is based on the Agile Manifesto, it will enable the tool to be generic, which means that it can be used to perform assessments on any Agile methodology, implementation or context.

The application shall be available to its user via World Wide Web and provide access to its users with user accounts. These accounts will enable users to operate different functions according to the account holder’s role on the agility assessment.

Notes and instructions for the users shall be included to guide the users during preparation for assessments, performing the assessments and gathering the results of the assessments.

Each assessment shall be recorded and made available to the related users for further analysis and modification purposes.

After completion of the assessments, the application shall enable users to get automated reports which will include graphical and commentary elements that will ease the understanding of the assessment results.

The users shall be able to compare assessments from their results and get automated reports for the comparison.

Different modes of usage shall be provided for assessments to be performed by single individuals and/or multiple individuals in teams.

Users shall be able to get assessment reports for different scopes such as project and organization.

The application shall be designed extensible to provide different agility assessment models to be adapted.

3.3 Product Perspective

AssessAgility is a web application that will be accessible via Web browsers. The application will be located on a cloud server and will run on a Windows Virtual Machine. The application itself is not a part or component of another software or system. However, it interacts with the users, web browsers and a cloud server. The context diagram of the system is shown in the figure below.

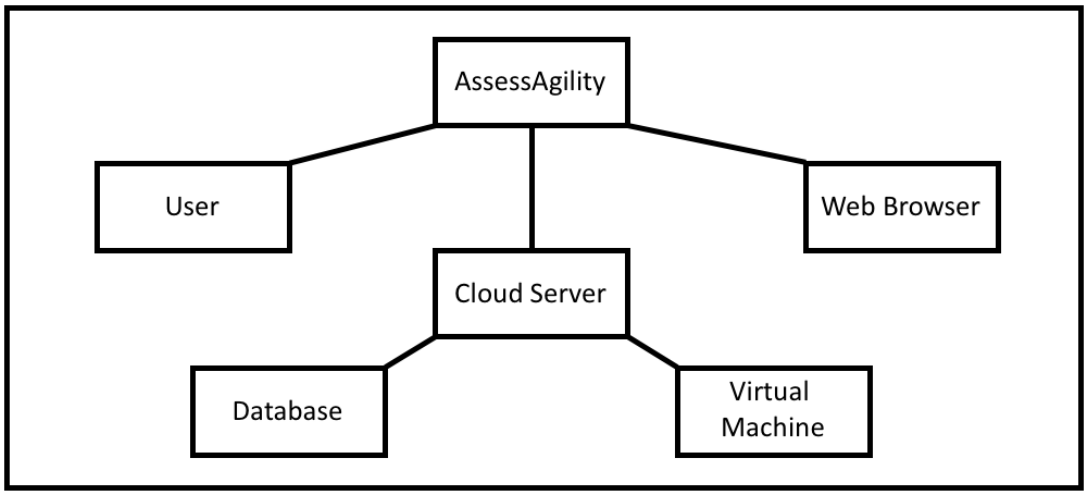


Figure 2: Jackson Context Diagram

3.3.1 User Interfaces

Those who are familiar with any kind of Web application, should be able to use the AssessAgility. The input devices for the application shall be mouse and keyboard of the client computers. Also, the application shall include support for the client devices with multi-touch screens. The gesture types supported for those devices shall be tap, flick, pinch and spread. Bootstrap [93] front-end web framework shall be used for developing a responsive user interface. The following figure demonstrates the GUI elements that will be employed on the systems user interface:

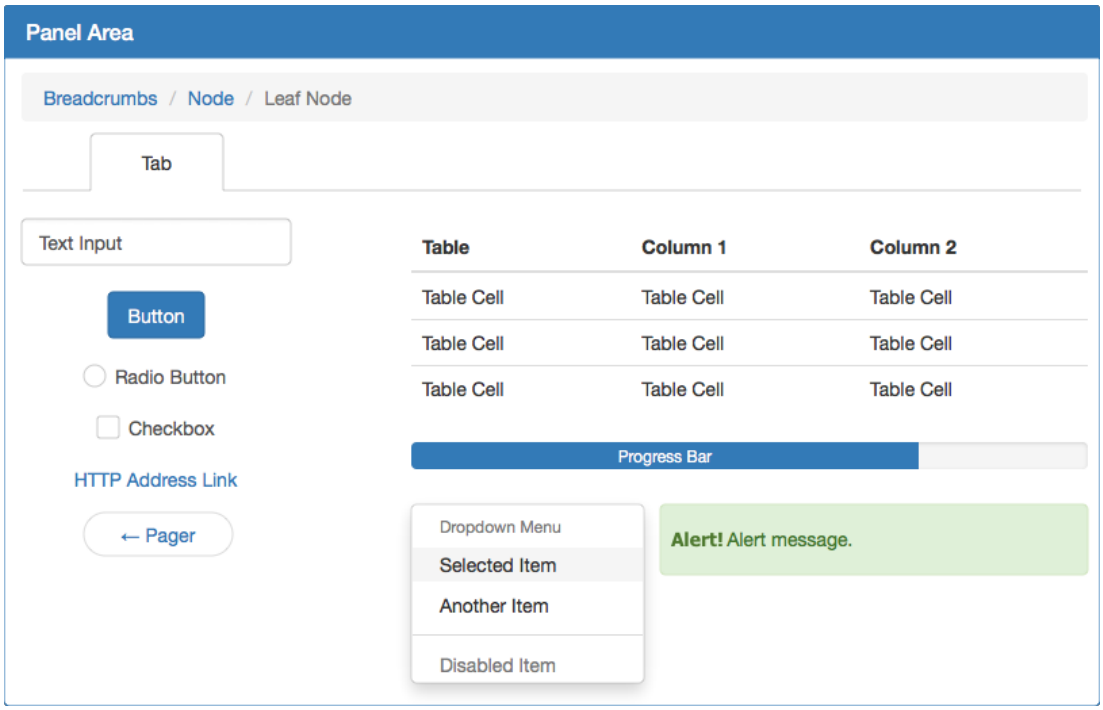


Figure 3: GUI Elements

3.3.2 Software Interfaces

The development language for the application will be C# and Microsoft Visual Studio will be used as an integrated development environment (IDE). Furthermore, Javascript, Bootstrap and SQL Server Compact (SQL CE) technologies will be used throughout the development process. The following table presents the software interfaces for the application.

Table 3: Software Interfaces

Interface Name	**Version No**	**Source**	**Purpose**	**Interface Definition**
Microsoft Azure	2016	Microsoft	Will be used to host, provide storage and backup for the application.	SaaS based cloud service.
Microsoft	2015	Microsoft	Will be used to develop the	Integrated Development

Interface Name	Version No	Source	Purpose	Interface Definition
Visual Studio			application.	Environment (IDE)
Bootstrap	3.3.7	GitHub	Will be used for the development of the Graphical User Interface (GUI).	Mobile first and responsive front-end framework
SQL CE	4.0	Microsoft	Will be used for database management.	Compact relational database

3.3.3 Communication Interfaces

The communication interfaces for the application shall be: HTTP Version 2.0 and TCP/IP.

3.4 Assumptions and Dependencies

- It is assumed that users of the AssessAgility have enough computer and information literacy to use the application.
- It is assumed that users of the AssessAgility are holding valid e-mail addresses.
- It is assumed that the client devices used to run the application have a Web browser installed.
- It is assumed that client devices have active Internet connections while running the application.

3.5 Product Functions

In this section, functions of the application are listed according to the user classes supported. AssessAgility includes 4 user classes: Admin, Lead Assessor, Assessor and Self-Assessor. The functions are categorized into user classes and each function is described as a separate use case that is written in the brief/summary format.

3.5.1 Generic User Functions

All the user classes present in the application are provided with login and profile management functions.

3.5.1.1 Login

User logins to the application by entering e-mail address and password that are stored in the database of the application.

3.5.1.2 Update Profile

User updates the information that is recorded in his/her profile.

3.5.2 Admin Functions

Admin user class performs administrative operations in the application. The functions provided to the Admin user class, includes management of organization and user entities in the application.

3.5.2.1 Manage Organization

Admin adds organizations to the application, edits the information of the existing organizations and deletes existing organizations from the application.

3.5.2.2 Manage User

Admin adds different types of users to the application, edits the information of the existing users and deletes the existing users from the system.

3.5.3 Lead Assessor Functions

Lead Assessor user class is responsible for the operations related to the management of assessments. The functions provided to the Lead Assessor user class includes management of project, team and assignment entities in the application. Also, functions such as analysis of assessments and report generation are also available to the Lead Assessor user class.

3.5.3.1 Manage Project

Lead Assessor adds projects to the application, edits the information of the existing projects and deletes the existing projects from the system.

3.5.3.2 Manage Team

Lead Assessor creates teams by adding members to them, edits the teams and deletes existing team formations from the application.

3.5.3.3 Manage Assignment

Lead Assessor makes assignments between projects and teams, edit the relations of the teams and projects within the assignments and undoes existing assignments.

3.5.3.4 Analyze Assessment

Lead Assessor analyzes the assessments performed by Assessor user class and approves or rejects the assessments to finalize them.

3.5.3.5 Generate Report

Lead Assessor obtains reports of the completed assessments. Three types of reports are available to Lead Assessor: single reports which presents the results of a single assessment, comparison reports: which presents results of multiple assessments in comparison and combination reports: which presents combination of assessment results for multiple assessments made by different assessors.

3.5.4 Assessor

Assessor user class is responsible for the filling of the assessments. The sole function available for Assessor user class is performance of assessments.

3.5.4.1 Perform Assessment

Assessor performs assessments assigned to them by Lead Assessor user class by gathering information about the practices presented in the assessment form.

3.5.5 Self-Assessor

Self-Assessor user class presents self-assessment takers who may have reach the application through search engines. The functions provided to Self-Assessor user class is an amalgam of the functions provided to the Lead Assessor and Assessor user classes. Therefore, in these terms, Self-Assessor user class inherits the functions Manage Project, Perform Assessment and Generate Reports from the Lead Assessor and Assessor user classes. On top of these functions, Self-Assessors are able to register to the application without the supervision of the Admin.

3.5.5.1 Manage Project

Self-Assessor adds projects to the application, edits the information of the existing projects and deletes the existing projects from the system.

3.5.5.2 Perform Assessment

Self-Assessor performs self-assessments on the projects that are previously added by him/her by gathering information about the practices presented in the assessment form.

3.5.5.3 Generate Report

Self-Assessor obtains reports of the completed assessments. Only single reports which presents the results of a single assessment are available for this user class.

3.5.5.4 Register

Self-Assessor registers to the application by providing a valid e-mail address and a password.

3.6 Specific Requirements

In this section, use cases are used to capture and establish functional requirements. The high-level use cases are given in the Use Case Diagram. (See the Figure 4 on the next page.) All the use cases are written in the Fully Dressed Format and at the User Goal Level. The table below shows the template used for stating use case scenarios:

Table 4: Template for Use Case Scenarios

Use Case:	<< Arbitrary use case number for reference purposes>>
Scope	<<Boundary of the system under design >>
Priority	<<Priority level of this use case >>

Description	<<The purpose of this use case >>
Primary Actor	<< The name of the principal actor that calls upon system services to fulfill a goal >>
Precondition(s)	<< Conditions that must be applied before the steps of main success scenario. >>
Main Success Scenario	<< Successful path of actions taken in this scenario >>
Extensions	<< Alternate steps that should be taken instead of the ones in the main success scenario >>
Post condition(s)	<< Conditions that occur after the steps of main success scenario >>
Exceptions	<<Steps that can be happened during exception >>
Special Requirements	<<Non-functional requirements, quality attributes, formulas, or constraints relates specifically to this use case >>
Reference	<< Use case diagram(s), user interfaces, and other entities related to the use case >>

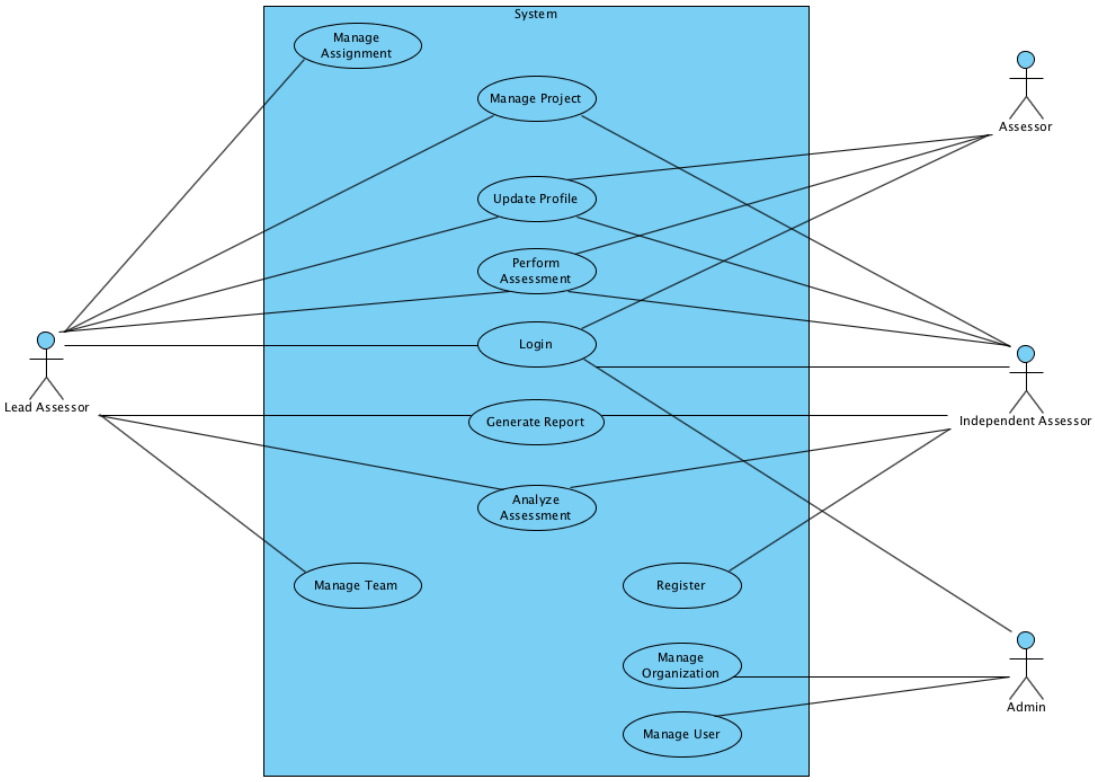


Figure 4: UCD Diagram

The use case scenarios of the AssessAgility are given at the Appendix A.

3.7 Logical Database Requirements

The Entity-Relationship Diagram of the AssessAgility is attached to the Appendix B.

3.8 Software Design Descriptions

This section includes two design viewpoints namely logical and interface for the AssessAgility. In logical viewpoint, the static structure (classes, data types) of the software is presented and indicated with UML class diagrams. In the interface viewpoint, the communication interfaces of the various components of the system are presented with UML component diagrams.

3.8.1 Logical Viewpoint

In this viewpoint, the static structure of the classes that make up the software product is designed with Visual Studio 2015 Class Designer functionality. The resulting class diagram depicts the hierarchical relationship between the classes and each class includes the properties, methods and the interfaces it provides to the other classes. The class diagram of the AssessAgility is attached to the Appendix B.

3.8.2 Interface Viewpoint

In this viewpoint, the communicating components of the system, are presented with UML Version 2.5 Component Diagram. The components are designed to achieve maintainability and modularity. Especially, the Assessment Model Component is designed to provide extensibility for different agility assessment models to be integrated to the system and it's based on the meta-model described in [58]. Furthermore, the MVC pattern is employed for the overall design. The component diagram of the AssessAgility is attached to the Appendix B.

3.8.3 Interaction Viewpoint

In this viewpoint, the the sequence of interactions between classes are given with the UML Version 2.5 System Sequence Diagrams. For this viewpoint, the sequence diagrams that describes the critical behavior of the system were drawn and trivial CRUDL type of interactions are omitted. The sequence diagrams of the AssessAgility is attached to the Appendix B.

CHAPTER 4

EXEMPLAR AGILITY ASSESSMENT PROCESS

This chapter presents an exemplar agility assessment process that extends the AgilityMOD: Agility Assessment Reference Model [84]. In the purpose and scope section, objectives and limitations of the exemplar assessment process are given. In the introduction to AgilityMOD: Agility Assessment Reference Model section, the structure and components of the reference model are explained briefly. In the roles and responsibilities section, the roles and their responsibilities, required for performing an assessment are explained. In the assessment process section, the phases, underlying processes, and the activities of the exemplar assessment process are explained with details.

4.1 Purpose and Scope

This exemplar agility assessment process provides rules and guidelines for conducting agility assessments with the AgilityMOD: Agility Assessment Reference Model. The process consists of set of instructions designed to guide and advise the assessment conductors.

The processes provided does not presume any specific Agile methodology, context or implementation. Hence, it is applicable for performing agility assessments in different types of application domains and sizes of organizations.

4.2 Introduction to AgilityMOD: Agility Assessment Reference Model

The AgilityMOD is based on the structure of the ISO/IEC 15504 Software Process Improvement and Capability Determination (SPICE) Model, Part 2 [83] and Part 5 [94] and it provides a well-defined structure to assess projects' and organizations' agility level and introducing roadmaps to organizations in adopting agile principles/practices.

The model, consists of two dimensions: the aspect dimension and the agility dimension as seen on the Figure 5. In the aspect dimension, aspects are defined as Exploration, Construction, Transition and Management which are derived from agile processes and practices. In these terms, aspects present special kinds of agile compatible process clusters. In the agility dimension, agility of an aspect is described with a-four-point ordinal scale which enables the agility to be assessed at "Not

Implemented", "Ad-Hoc", "Lean" and "Effective" levels. When an aspect progresses from the bottom level: "Not Implemented" to the top level: "Effective", its conformance to agile values and principles increases.

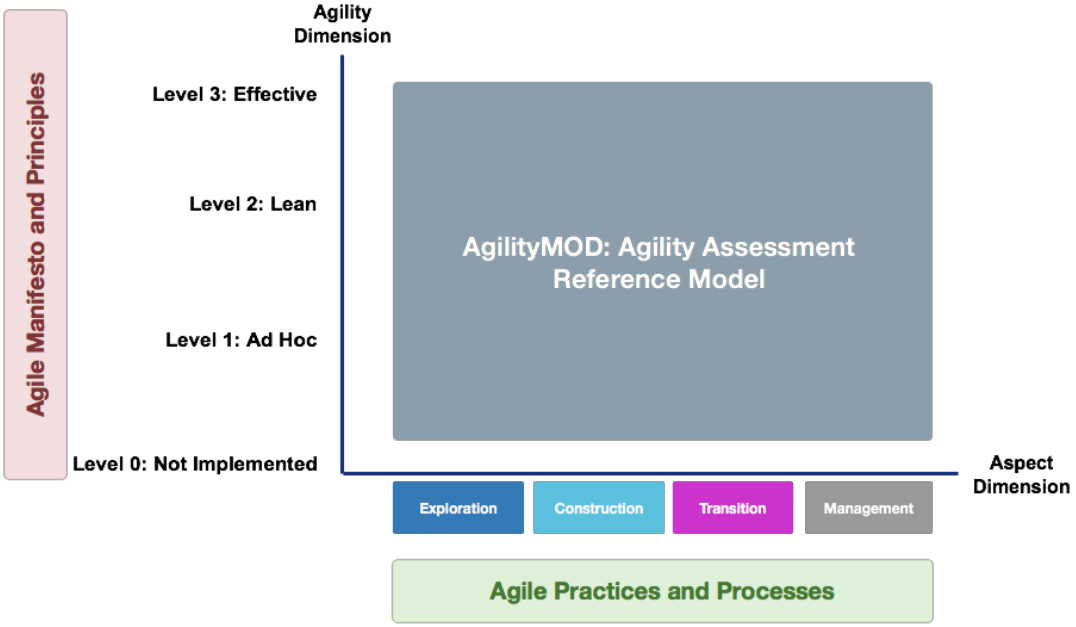


Figure 5: Dimensions of AgilityMOD

In the model, assessment of agility is performed based on the assessment indicators which are aspect practices and work products for aspect dimension, and generic agility practices, generic resources and generic work products for agility attributes at agility dimension [85].

Assessment is performed through the aspect attributes that belongs to agility dimension of the model. During the assessment, aspect practices and generic practices belonging to the aspects are rated based on the achievement. The rating approach described in the ISO/IEC 15504-Part 2 is adopted for the model and rating is based on a four-point ordinal scale:

- Not Achieved (0-15% achievement percentage)
- Partially Achieved (16%-50% achievement percentage)
- Largely Achieved (51%-85% achievement percentage)
- Fully Achieved (86%-100% achievement percentage)

For more information on the reference model, [84] could be visited.

4.3 Roles and Responsibilities

4.3.1 Assessment Sponsor

Assessment Sponsor is the individual that represents the organization unit that requires the assessment. Assessment Sponsor provides sponsorship, provides information about business needs and objectives, and approves the Assessment Plan. The responsibilities of the Assessment Sponsor role are given below:

- Defines the organizational unit to be assessed
- Acts as an interface between upper management and spreads the value and impact of the Agility assessment and improvement
- Ensures that budget and other supporting factors are provided
- Communicates with Lead Assessor about business objectives, needs and other assessment factors
- Reviews and approves Assessment Plan
- Reviews and receives the Assessment Report

4.3.2 Lead Assessor

Lead Assessor holds the responsibility for the assessment process and the assessment team. Lead Assessors manage, coordinate the assessment process and acts as an interface between Assessment Sponsor and the Assessment Team. The responsibilities of the Lead Assessor role are given below:

- Meet with the Assessment Sponsor to obtain information on organizational unit, business objectives, needs and other assessment factor before the assessment
- Create Assessment Plan by cooperating with the Assessment Sponsor
- Get Approval for the Assessment Plan
- Create team and assign team roles
- Obtain commitment for the assessment
- Prepare team members for the assessment by providing information on appraisal process and schedule
- Resolve conflicts and disagreements
- Track and monitors schedule and performance
- Ensure that the assessment process is followed
- Verify and validate data

- Derive aspect levels based on the data and aspect practice ratings
- Create and review Assessment Report
- Deliver Assessment Report to the Assessment Sponsor

4.3.3 Assessor – Assessment Team Member

Assessors are experts who perform the interviews with assessment participants and gather objective evidence within the interviews. Usually, assessors are part of an assessment team that consists of 2-3 assessors. The responsibilities of the Assessor role are given below:

- Commit to Assessment Plan
- Perform interviews with the assessment participants
- Gather objective evidence
- Review the objective evidence according to the aspects and aspect practices
- Record objective evidence according to the aspect practices
- Take and record supplementary notes
- Reconcile objective evidence and notes with other assessors on the team
- Derive aspect practice ratings based on the objective evidence and supplementary notes
- Deliver records and ratings

4.3.4 Assessment Participants

Assessment participants are members of the organizational unit which is responsible for the project that is selected for the agility assessment. Their main purpose is to provide information during interview about the related aspects and aspect practices. The responsibilities of the Assessment Participant role are given below:

- Commit to Assessment Plan
- Attend to interview sessions
- Provide information and artifacts about the related aspects and aspect practices

4.4 Assessment Process

The assessments are conducted by an assessment team whose member(s) are independent of the Organizational Unit being assessed. An independent assessment may be conducted, for example, by an organization on its own behalf as independent

verification that its assessment program is functioning properly; the assessment sponsor will belong to the same organization but not necessarily to the Organizational Unit being assessed [29].

The table below presents the overall description of the assessment process:

Table 5: Overall Description of the Assessment Process

Phase	**Process**	**Purpose**	**Activities**
Planning	Analyze Requirements	Understand the business needs for the organizational unit that requested the assessment.	1. Determine Assessment Objectives 2. Determine the Project 3. Determine Data Collection Strategy 4. Determine Assessment Constraints 5. Determine Assessment Scope 6. Determine Assessment Outputs 7. Determine Assessment Activities
	Identify Assessment Resources	Identify and estimate the required resources for the assessment.	1. Identify Team Members 2. Identify Assessment Participants 3. Identify Equipment and Facilities
	Create	Generate an	1. Determine

	Assessment Plan	assessment plan based on the requirements and resource estimations.	Assessment Activities 2. Develop the Assessment Cost and Schedule 3. Generate Assessment Plan
	Prepare for Assessment	Ensure schedule availability and participant and assessment team commitment.	1. Obtain commitment for Assessment
Data Collection	Perform Assessment	Gather objective evidence against each attribute under each aspect.	1. Interview Participants 2. Record and Examine Objective Evidence 3. Assign Aspect Practice Ratings
Validation and Agility Level Determination	Validate the Assessment Results and Determine Agility Levels	Validate the evidence gathered to form objective results of the assessment. Derive levels based on the validated evidence.	1. Verify and Validate Data 2. Determine Agility Levels
Reporting	Generate and Deliver the Assessment Report	Document and present the assessment results.	1. Generate Assessment Report 2. Deliver the Assessment Report

The figure below demonstrates the main processes of the assessment process.

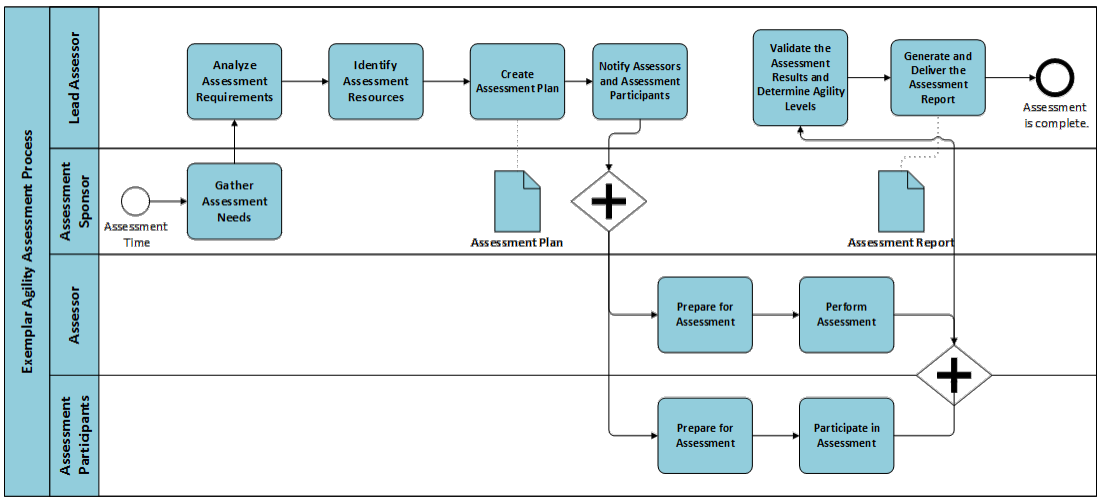


Figure 6: Exemplar Agility Assessment Process Diagram

4.4.1 Planning Phase

Planning phase ensures that the preliminary thinking and work that will define and state the purpose and the scope of the assessment are made. Planning phase includes analysis of the requirements for the assessment, identification of the project that the assessment will be performed on, identification and selection of the assessment participants, creation of the assessment team and preparation for the assessment conduct. The processes included in the planning phase are given in details below.

4.4.1.1 Analyze Assessment Requirements

Table 6: Analyze Assessment Requirements Process Definition

Purpose	Understand the business needs of the organization unit that requested the assessment. Lead Assessor collects the information from the Assessment Sponsor and then define the purpose for the assessment which is aligned with the business needs and objectives. Discuss and define the objective evidence collection strategy. The Lead Assessor collaborates with the Assessment Sponsor to form the basis of the data collection strategy.
Entry Criteria	• Assessment Sponsor has decided that an Agility Assessment is required. • Assessment Sponsor and other organizational unit members are available for providing business and assessment needs and objectives.
Inputs	• Business needs and objectives

	• Assessment requirements and constraints
Activities	• Determine Assessment Objectives • Determine the Project • Determine Data Collection Strategy • Determine Assessment Constraints • Determine Assessment Scope • Determine Assessment Outputs
Outputs	Assessment Requirements Specification
Exit Criteria	The assessment purpose, scope and constraints are defined. Initial data collection strategy is developed and recorded to the assessment plan.

Activities

Table 7: Analyze Assessment Requirements Process Activities

No.	Activity	Description
1	Determine Assessment Objectives	Lead Assessor communicates with Assessment Sponsor to gather and record the objectives of the assessment which are aligned with the organizational unit's business needs and objectives.
2	Determine the Project	The project which reflects the organization unit's needs and objectives most, is selected for the assessment. The information of the selected project is recorded.
3	Determine Data Collection Strategy	Data collection strategy relies highly on the assessment objectives. Lead Assessor and Assessment Sponsor communicate and define a strategy that fits the assessment objectives.
4	Determine Assessment Constraints	The constraints such as the resources, schedule and cost are important factors for the conduct of the assessment. Lead

		Assessor and Assessment Sponsor considers and negotiates these constraints that affect the assessment.
5	Determine Assessment Scope	The assessment scope consists of the scope of the AgilityMOD Reference Model and scope of the project that is subject to the assessment. Determination of the scope is done to conduct the assessment.
6	Determine Assessment Outputs	Determination of the specific outputs of the assessment such as aspect practice ratings and achieved agility levels of aspects.

4.4.1.2 Identify Assessment Resources

Table 8: Identify Assessment Resources Process Definition

Purpose	Identify and estimate the required resources for the assessment. These resources may include participants for the assessment, facilities, repositories and other information sources.
Entry Criteria	Assessment requirements are defined.
Inputs	Assessment Requirements Specification
Activities	• Identify Team Members • Identify Assessment Participants • Identify Equipment and Facilities
Outputs	Initial Assessment Plan
Exit Criteria	The required information and facility resources for the assessment conduct are identified and recorded.

Activities

Table 9: Identify Assessment Resources Process Activities

No.	Activity	Description
1	Identify Team Members	The identity and responsibilities of the team members who will conduct the assessment are identified. Ensure that a qualified team is available and ready to perform the

		assessment.
2	Identify Assessment Participants	The identity and responsibilities of the assessment participants in the assessments are determined.
3	Identify Equipment and Facilities	The specific equipment such as voice recorders, projectors and computers, and facilities such as meeting rooms are identified.

4.4.1.3 Create Assessment Plan

Table 10: Create Assessment Plan Process Definition

Purpose	Identify and estimate the required resources for the assessment. These resources may include participants for the assessment, facilities, repositories and other information sources.
Entry Criteria	Assessment requirements are defined.
Inputs	Assessment Requirements Specification
Activities	• Determine Assessment Activities • Estimate the Assessment Cost and Schedule • Create Assessment Plan
Outputs	Initial Assessment Plan
Exit Criteria	The required information and facility resources for the assessment conduct are identified and recorded.

Activities

Table 11: Create Assessment Plan Process Activities

No.	Activity	Description
1	Determine Assessment Activities	The activities to be performed in conducting the assessment are defined to allocate resources and schedule for these activities.

2	Estimate the Assessment Cost and Schedule	The activities to be performed in conducting the assessment are defined to allocate resources and schedule for these activities.
3	Create Assessment Plan	Assessment plan is created which includes at minimum: • Description of the assessment objectives • Description of the project selected • Description of the data collection strategy • Description of the assessment constraints • Description of the assessment scope • Description of the assessment outputs • Activities to be perform during assessment The resources and schedule assigned to these activities

4.4.1.4 Prepare for Assessment

Table 12: Prepare for Assessment Process Definition

Purpose	Ensure schedule and resource availability. Lead Assessor obtains participant and assessment team commitment. Provide information to participants for them to become familiar with the assessment structure and context.
Entry Criteria	Assessment Plan is created.
Inputs	Assessment Plan
Activities	Obtain commitment for Assessment
Outputs	Agreed Assessment Plan
Exit Criteria	Commitment to the Assessment Plan is obtained.

Activities

Table 13: Prepare for Assessment Process Activities

No.	Activity	Description
1	Obtain commitment for Assessment	Assessment plan is reviewed and agreed by the Lead Assessor and Assessment Sponsor. The plan is supplied to the participants and team members and their commitment to the plan is obtained.

4.4.2 Data Collection Phase

4.4.2.1 Perform Assessment

Table 14: Perform Assessment Process Definition

Purpose	Obtain information about the specific details about the implementation of aspect and generic practices for each of the aspects. Identify and record potential issues, improvement suggestions, weaknesses and strengths.
Entry Criteria	• Assessment Plan is created. • Commitment for the Assessment is obtained. • Assessment participants are present.
Inputs	Assessment Plan
Activities	• Interview Participants • Record and Examine Objective Evidence • Assign Aspect Practice Ratings
Outputs	• Interview records • Objective Evidence • Aspect Practice Ratings
Exit Criteria	Objective evidence on each aspect and generic practice is collected and aspect practice ratings are given.

Activities

Table 15: Perform Assessment Process Activities

No.	Activity	Description
1	Interview Participants	Interview and ask questions about the implementation of aspect and generic practices for each of the aspects. Take detailed notes to capture the responses. If required provide definitions about the components of the assessment such as aspects and practices. Interview each participant separately and cross validate the interview findings.
2	Record and Examine Objective Evidence	Record interview findings about current implementation of the practices and potential issues, improvement suggestions, weaknesses and strengths. Evaluate the evidence against the reference model to determine the evidence's appropriateness and adequateness for the implementation of model practices. Take corrective actions and if required repeat the interviews.
3	Assign Aspect Practice Ratings	Relate the gathered information for each practice and give ratings to aspect practices based on the objective evidence recorded.

4.4.3 Validation and Agility Level Determination Phase

4.4.3.1 Validate the Assessment Results and Determine Agility Levels

Table 16: Validate the Assessment Results and Determine Agility Levels Process Definition

Purpose	Validate the correlation between data collected and aspect practice ratings. Derive aspect attribute ratings from the aspect practice ratings.
Entry Criteria	• Data collection is completed.
Inputs	• Interview records • Objective Evidence • Aspect Practice Ratings

Activities	• Verify and Validate Date • Determine Agility Levels
Outputs	Aspect Agility Levels
Exit Criteria	Correlation between data collected and aspect practice ratings are adequate. Agility Levels of each aspect is determined.

Activities

Table 17: Validate the Assessment Results and Determine Agility Levels Process Activities

No.	Activity	Description
1	Verify and Validate Date	Verify and validate the data collected to: • Ensure that the data is objective and consistent as a whole, • The data is sufficient and relevant to the purpose and scope of the assessment • The data represents the related aspect that it's collected for If required held feedback sessions to get affirmation from the assessment participants and compare the feedbacks of each participant.
2	Determine Agility Levels	Analyze aspect practice ratings and derive an agility level for each aspect. Record the determined level.

4.4.4 Reporting Phase

4.4.4.1 Generate and Deliver Assessment Report

Table 18: Generate and Deliver Assessment Report

Purpose	Generate and deliver the assessment report including all the collected data, strengths, weaknesses, improvement suggestions, aspect practice ratings and agility levels.

Entry Criteria	• Data collection is completed. • Validation and Agility Level Determination is completed.
Inputs	• Interview records • Objective Evidence • Aspect Practice Ratings • Aspect Agility Levels • Assessment Plan
Activities	• Generate Assessment Report • Deliver Assessment Report
Outputs	Assessment Report
Exit Criteria	The assessment purpose, scope and constraints are defined. Initial data collection strategy is developed.

Activities

Table 19: Generate and Deliver Assessment Report Process Activities

No.	Activity	Description
1	Generate Assessment Report	Prepare an assessment report that includes collected data, strengths, weaknesses, improvement suggestions, aspect practice ratings and agility levels in a format that is comprehensible to guide the process improvement and decision making processes.
2	Deliver Assessment Report	Deliver the assessment report to the Assessment Sponsor.

4.5 Evaluation of the Exemplar Agility Assessment Process

For the evaluation of the Exemplar Agility Assessment Process, expert judgement was utilized. The final version of the process presented in this chapter was developed iteratively and shaped by the opinions of three experts. After each release, the process was put into a review process and suggestions and opinions of the experts were gathered. Then, the suggestions and opinions are discussed in meetings were all the experts were present and mutually agreed changes were implemented to the process.

CHAPTER 5

APPLICATION OF ASSESSAGILITY

This chapter presents application of AssessAgility in a multiple case study setting that is conducted for the validation of the tool. In the multiple case study section, the details of the multiple case study including the design, conduct, and the findings are given.

5.1 Multiple Case Study

After the development of the AssessAgility, we have conducted a multiple case study for the validation of the tool. The aim of this case study is to determine whether or not the tool meets the expected criteria we have defined, is able to guide and automate the assessment process as a whole and is able to provide efficient usage for performing agility assessments.

In the light of the objectives given above we have defined the following research questions:

RQ1: To what extent is the tool sufficient to meet the expected criteria determined after the literature review of this study?

RQ2: To what extent is the tool able to automate and guide the assessment process?

RQ3: How efficient is the tool for the assessment process?

RQ4: What are the strengths and weaknesses of the tool?

5.1.1 Design of the Multiple Case Study

Case Selection Strategy:

To increase the reliability of our validation, we plan to conduct a multiple case study including three cases. Each case study will be conducted on a different project from different business domains and from different types and sizes of organizations. The only strict requirement for projects is that they should be carried out in an Agile setting. In the scope of the case studies we plan to conduct an agility assessment covering 4 aspects defined in the reference model for each project. Each assessment will be performed in conformance to the exemplar assessment process defined in the Chapter 4. To perform the assessments, we aim to form an assessment team that consists of two assessors and one assessment leader. These experts will be assigned

to relevant roles, according to their experience in the field. We also plan to pick software improvement experts which are not part of this thesis study to increase objectivity.

Data Collection Strategy:

In the *Data Collection* phase of the assessment process we plan to conduct structured interviews with the aspect owners of the projects. The aspect owners that are planned to be interviewed include at least one requirements engineer, one product owner, one developer, one tester and one configuration manager. The main reason behind conducting interviews with different roles is to get direct responses from relevant practitioners. The responses given in the interviews will be recorded to the related fields provided in the tool and will be recorded at a database located in a cloud server.

Validation Strategy:

After the end of the *Validation and Agility Level Determination* and *Reporting* phases of the assessment process are over, the generated reports will be discussed with the aspect owners to discuss about the results of the assessment.

5.1.2 Conduct of the Multiple Case Study

As stated in the case study design we performed three case studies on three projects in different business domains and from different types and sizes of organizations. An assessment team is formed with an assessment leader who has 5 years of experience as an SPI expert and two assessors one of them having one and a half years of experience and the other one having one year of experience as an SPI consultant. Before starting to conduct the case studies, each team member in the assessment team completed a manual assessment without the tool to be able to grasp the reference model and the exemplar assessment process. The total time for each assessor to complete the manual assessment is tracked and recorded to compare with the assessments performed with the tool. The team followed the steps of the exemplar assessment process given in the Chapter 4 by using the tool. Then, an assessment report for each case is obtained and discussed with the assessment participants and their teams for validation. The details and findings of each case are given in the next section below.

After the interviews, we also wanted assessors to answer the following questions.

Table 20: Case Study Evaluation Form

ID	Question	Related RQ
Q1	Please rate the tool with the evaluation questionnaire provided.	RQ1
Q2	Does the tool cover the assessment process as a whole? If not, please describe the missing parts of the process.	RQ2

ID	Question	Related RQ
Q3	Does the tool guide the assessment process as a whole? If you required additional guidance about the assessment process, please describe the parts where you needed guidance?	RQ2
Q4	Please rate the usability of the tool with the questionnaire provided.	RQ3
Q5	Would you prefer performing agility assessments with the tool or without the tool? Why?	RQ3
Q6	Does the tool reduce the effort spent for performing assessments? Could you please describe the difference?	RQ3
Q7	Could you please describe the strengths and weaknesses of the tool?	RQ4

For questions 1 and 4 we want assessors to fill the questionnaires provided in the Appendix C and D. For question 1 we have used Evaluation Questionnaire, for question 2 we have used USE Questionnaire: Usefulness, Satisfaction, and Ease of Use based on [95].

5.1.3 Findings of the Multiple Case Study

In this section, first we provided information about the case studies and then we gave the assessment results of each case. The information about the projects and organizations are kept secret for confidentiality purposes.

5.1.3.1 Case Study 1

In the scope of the Case Study 1, the assessment is performed through a total of three-hour interview session with two team leaders. Each team leader is interviewed by one assessor and whole interview is completed within total of 12 person-hours. The details of the Case Study 1 are given in the table below:

Table 21: Information about Case Study 1

Organization ID:	Organization GS
Size of the Organization:	Medium SME (100-250 Employees)
Agile Experience:	>3 Years
Organization Bio:	Organization GS is data analytics service provider which offers various SAAS based web applications for its customers. Currently it has 20 customers constantly using their various services such as data visualization, data processing, decision

	support, social network analysis, marketing analytics and customer behavior analysis. The organization decided to transit to Agile about four years ago with the main objectives of reducing time to market and enhancing software maintainability. All the current projects are applying Agile and Scrum is used throughout the projects with some customizations.
Project ID:	Project 1
Project Bio:	The project assessed in the scope of the study was a new development project which involves development of an SNA Module for an existing web service framework. It has several interfaces to other modules such as data visualization, forecasting and prediction. The framework which the developed SNA Module will be a part of is a group of web application services which provides continuous service delivery, data protection and prediction.
Project Team Bio:	The project was carried out by 2 separate teams with total of 20 team members. The distribution of the roles is given below: • 2 Team Leaders • 7 Software Developers • 2 Software Architects • 2 Data Visualization Analysts • 5 Data Analysts • 2 Testers

Assessment Results:

We assessed the Organization GS over the Project 1 which is a new development project that involves development of an SNA Module for an existing web service framework. The achieved agility levels for the project can be seen on the Figure 7 below:

Figure 7: Achieved Agility Levels of Project 1

The results show that all aspects are at the Level 3: Effective which means that Agile engineering methods/practices are internalized, tools are integrated to aspects, and learning and improvement are in place [84]. The ratings of the all aspect practices are given at the Figure 8 below:

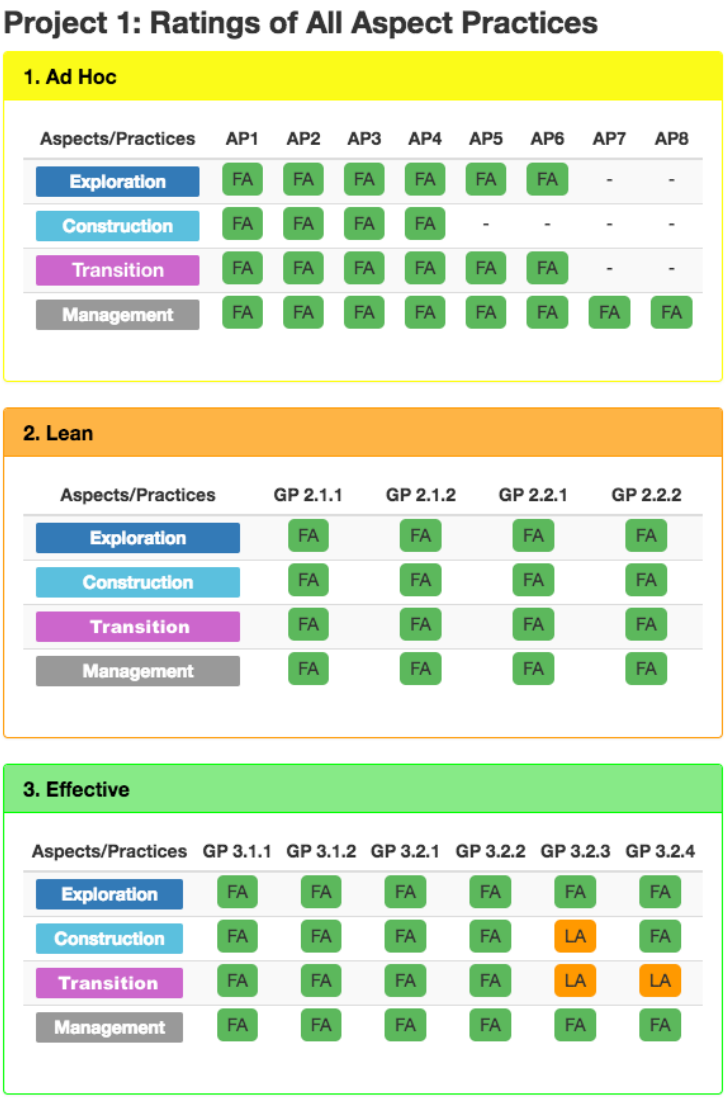


Figure 8: Ratings of All Aspect Practices of Project 1

The assessment report which is generated automatically with the tool, is given at the Appendix F. The aspect based findings revealed through the assessment are given below:

Exploration Aspect - Project 1:

According to the assessment results of the Project 1, the Exploration Aspect is at Level 3 – Effective. Every practice in this aspect is rated as fully achieved (FA). Below the findings, strengths, weaknesses and improvement suggestions related to each level are given.

Exploration Aspect – Level 1 – AdHoc

For the projects in the Organization GS, the business Analysis department defines the business needs for the modules. Usually customers select an existing solution based on their needs, however when a customer need cannot be met with the current asset base, the business analysts record that customer need into Jira. The recorded customer needs are discussed in meetings with attendance of module owners and CEO. If a customer need is decided for implementation, the business analysts turn it into a detailed business need and add it to the Jira. Then the business needs recorded in the Jira are elaborated by the related modules' teams, the business analysts, and the customer. These business needs are turned into detailed user stories and again recorded into Jira. Dependencies between customer needs, business needs and user stories are also specified in the Jira tool.

Finally, the detailed user stories are put into the backlog according to their business values. The backlogs are groomed with adjustment meetings that are hold weekly. When a change request is received, it is recorded to the backlog, but is not integrated to the current sprints and discussed later. All changes are assessed in terms of potential risks and then impact analysis is performed.

Exploration Aspect – Level 2 – Lean

For effective communication, team members get together in daily stand-up meetings. Communication channels for both internal and external stakeholders are in place and all stakeholders have access to project related artifacts (backlogs, meeting logs, issue lists). Furthermore, the customer is explicitly integrated to the development process with regular on-site meetings. To balance the predictive work and adaptive work, weekly adjustments at product backlog are done. These adjustments enable a balanced work flow by keeping the items going in and out to sprints in control. To minimize the ceremony, checklists prepared from acceptance criteria recorded on the user stories, are being used to review the products. Also, retrospective meetings are held at the end of each sprint to gather feedback from the team. To reduce unnecessary documentation, document templates are designed with include-if-crucial mindset and these templates are used by teams. Producing unnecessary documentation or work is given a penalty according to the organization's culture.

Exploration Aspect – Level 3 – Effective

As for Agile engineering practices and methods, TDD is applied through acceptance criteria recorded in the user stories. Also, team members perform pair programming. Jira is integrated to the exploration aspect for management of product and sprint backlogs and a portal is used to access to other documents. For collaboration and shared responsibility, team members collaboratively perform project related work and decision making is also done collectively. Furthermore, all members share responsibility for Exploration artifacts, except the situations where domain related knowledge is required. Team Leaders have adopted Agile leadership styles. The team leader is seen as a representative of the team and the role is switched between team members. For encouragement of learning, teaching and improvement, teams in the organization continuously improve themselves with retrospective meetings and

peer to peer learning mechanism is promoted and employed between team members. To facilitate learning and improvement through measures, organization defined custom metrics to track and monitor the Exploration activities such as backlog velocity. The definitions of these metrics are kept in portal and tools are used to collect the metrics.

Construction Aspect - Project 1:

The assessment results showed that the Construction Aspect of the Project 1 is at Level 3 – Effective. 13 out of 14 practice in this aspect are rated as fully achieved (FA) and one practice is rated as largely achieved (LA). The findings, strengths, weaknesses and improvement suggestions for each level are given below.

Construction Aspect – Level 1 – AdHoc

To elaborate the work items such as the user stories defined in the exploration activities, sequence diagrams are drawn. For design exploration, architects in the team develop design solutions that are discussed and evaluated in terms of the functional and quality requirements. Class and sequence diagrams are drawn to explore the designs. For development C# and Java programming languages are used. A coding standard which includes rules on commenting is applied in the project. To ensure the correctness of the software at developer level, developers employ automated unit tests. Also, pair reviews are employed to review the code for refactoring.

Construction Aspect – Level 2 – Lean

Software is developed in an iterative and incremental fashion through sprints and the team works in a shared space and communicate through daily stand-up meetings and natural communication channels. Predictive and adaptive work for is balanced through limiting WIP and backlog adjustments. Frequent demos are made to check the solutions. Management of the design and coding activities are performed informally with peer reviews. Dependencies between design elements are stored on Jira with exported diagrams.

Construction Aspect – Level 3 – Effective

INVEST criteria are being used for requirements. Planning Poker technique is being employed for estimation. Automated unit and acceptance tests are applied. Peer reviews are held to validate the code against coding standard and code is refactored after the reviews. SourceTree tool is used for configuration management and check out and check in mechanisms for code. Also, SmartBear tool is used for code reviews.

Team members select their tasks voluntarily and collaborate during development. Some specialty required tasks are handled by specific team members such as data analysts or visualists. When a problem occurs, team resolves it collectively and the cause of the problem is investigated and required cautions are taken to avoid problems occurring again.

Team has a special directory on portal for learning. Team members share resources and record the retrospectives to that directory which is open to everyone. Metrics about code are kept on SmartBear tool. Defect density and review code coverage are example metrics that are collected.

- There is not a specific training mechanism available to address the deficiencies of the team members on Agile related topics. Therefore, to address these deficiencies trainings can be acquired.

- The metrics that are used by the team are products of GQM effort which is completed a year ago and after its completion the strategy itself and the accompanying techniques were never updated or improved. Hence, the metric warehouse should be analyzed and trimmed to keep the metric collection aligned with the changing business goals.

Transition Aspect - Project 1:

The Project 1 is achieved Level 3 – Effective in the Transition Aspect. Out of 16 practices two of them received largely achieved (LA) rating and remaining ones achieved fully achieved (FA) rating. The level based findings, strengths, weaknesses and improvement suggestions are given below:

Transition Aspect – Level 1 – AdHoc

The team keeps all the coding related artifacts under configuration control and all are artifacts are kept up to date. Also, each artifact is put into version control and check-in-check-out mechanism is used. Continuous integration is done daily and integrated code is put into the open environment. To ensure continuous integration, builds and deployments are done automatically. Automated tests are run to check the correctness of the deployed code after each deployment and customers are given access to the deployment area. Test scenarios are written according to the acceptance criteria stored in the user stories. Also, non-functional attributes of the deployments are tested. After the initial tests defects are recorded and before deployment, there's a regression and acceptance test process. Whole process is made visible via SourceTree tool. The supporting documentation criteria are specified in the organization level. Specific documents are created with the goal of improving understandability of the solutions.

Transition Aspect – Level 2 – Lean

Transition activities are performed in an iterative and incremental way. The team works in a shared space and communicate through daily stand-up meetings and natural communication channels. As for Agile practices and methods, Test Driven Development is integrated into the transition aspect. Test cases and codes are being in development with the same time as the code. Non-value added activities are eliminated and decision making is being made informally.

Transition Aspect – Level 3 – Effective

The team is self-organized and team member share the responsibility. When a problem occurs, team resolves it collectively and the cause of the problem is investigated and required cautions are taken to avoid problems occurring again. The team has a special directory on the portal for learning. Team members share resources and record the retrospectives to that directory which is open to everyone. The defects found are stored with the information such as phase injection and root cause. Defect, build and deployment statistics are collected.

- Some team members specialized on specific areas based on their previous experience. Periodic role dispersion between team members can be suggested.

- There is not a specific training mechanism to address the deficiencies of the team members on Agile related topics. Therefore, to address these deficiencies trainings can be acquired.

- The metrics that are used by the team are products of GQM effort which is completed a year ago and after its completion the strategy itself and the accompanying techniques were never updated or improved. Hence, the metric warehouse should be analyzed and trimmed to keep the metric collection aligned with the changing business goals.

Management Aspect - Project 1:

In terms of the Management Aspect, the Project 1 achieved Level 3 – Effective. 18 out of 18 practices are rated as fully achieved (FA) and the findings, strengths, weaknesses and improvement suggestions are listed below:

Management Aspect – Level 1 – AdHoc

The Organization GS uses a module based architecture for its framework and each module in the framework is decided and developed according to the feasibility studies made. 2-page module bios are prepared which include vision and scope of the modules. Team members are allocated to the teams according to their experiences and expertize. Each team works on a specific module. Agile is the de facto development method for the team and team members educate themselves regularly on Agile values and practices. External stakeholders are aligned with the Agile approach of the organization through discussions made on project initiations.

Offices allocated to teams have open and private spaces to facilitate both communication and privacy. Historical data is used to make plans within the early sprint planning. The previous backlogs are used as proxies to estimate the effort and time (in terms of sprints) needed to develop new modules. Plans are structured according to the business value so high-value items are prioritized. The team utilizes historical data and proxy based estimation (PROBE) to make estimates. The progress

of the project is tracked internally by team leaders and externally by the process team. The results of the both internal and external monitoring are shared with teams. Project risks are generated by the module owners. Then the risks are approved, prioritized and tracked. Risk mitigation strategies are discussed with all stakeholders and corrective actions are taken.

Management Aspect – Level 2 – Lean

Monitoring and tracking activities are handled iteratively and incrementally. Daily Stand-up meetings and on-site meetings with customer facilitates effective communication. Plans and estimations are made and tailored continuously. Management activities are done informally without supervision and the team is self-organizing.

Management Aspect – Level 3 – Effective

Plans and estimations are made continuously and updated as more information is obtained through the progress. The Jira tool is used for management related aspect practices. Plans and estimates are made collaboratively by the team members. The team solves its own problems by quickly investigating and developing solutions. Also, team has a special directory on portal for learning. Team members share resources and record the retrospectives to that directory which is open to everyone. Progress is tracked via variance measures for the estimations and actual values to learn the estimation mistakes and improve the planning accuracy.

5.1.3.2 Case Study 2

The assessment for the Case Study 2 is performed through a total of five-hour interview session with a business analyst and a software developer. Each person is interviewed by one assessor and whole interview is completed within total of 15 person-hours. The details of the Case Study 2 are given in the table below:

Table 22: Information about Case Study 2

Organization ID:	Organization TX
Size of the Organization:	Small SME (25-100 Employees)
Agile Experience:	1,5 Years
Organization Bio:	Organization TX is an IT firm that mostly provides customizable IOT services for on-site business users. Their customer portfolio mainly consists of rent a car firms, event holders, fair and conference venues, universities, shopping malls, art galleries and museums. Their current services include indoor localization services that are based on iBeacons, vehicle tracking and remote device management.

	The organization decided to adopt Agile methodologies 2 years ago and since then they are applying a Scrum XP Hybrid for all of their projects.
Project ID:	Project 2
Project Bio:	The project assessed in the scope of the study was an existing solution implementation case which involved usage of an existing platform. It involved implementation of remote device management and communication services to a manufacturing factory.
Project Team Bio:	The project was carried out by a team that consists of 11 members. The distribution of the roles is given below: • 2 Business Analysts • 1 Business Owner • 2 Electrical Engineers • 5 Software Developers • 1 Tester

Assessment Results:

We assessed the Organization TX over the Project 2 which is an existing solution implementation case that involved usage of an existing platform. The achieved agility levels for the project can be seen on the Figure 9 below:

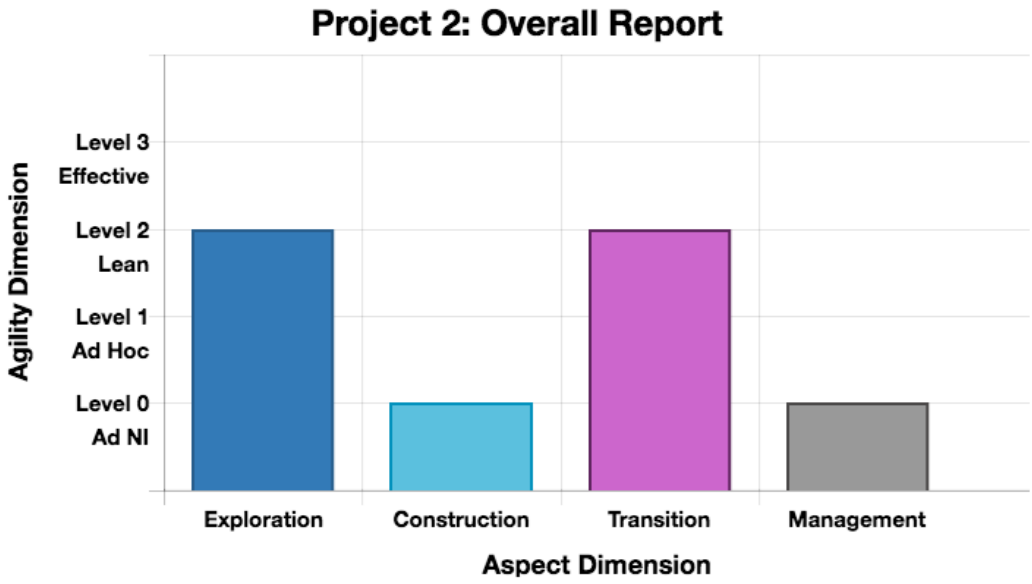


Figure 9: Achieved Agility Levels of Project 2

The results show that Exploration and Transition Aspects are at the Level 2: Lean which means that the organization is iterative and incremental, communication is effective, balance is achieved, and ceremony is minimized [84]. Construction and Management Aspects are at Level: 1 Ad-Hoc which means that Aspect Practices are achieved, transition attempts to Agile are present, and inconsistencies are present in Agile applications [84]. The ratings of the all aspect practices are given at the Figure 10 below:

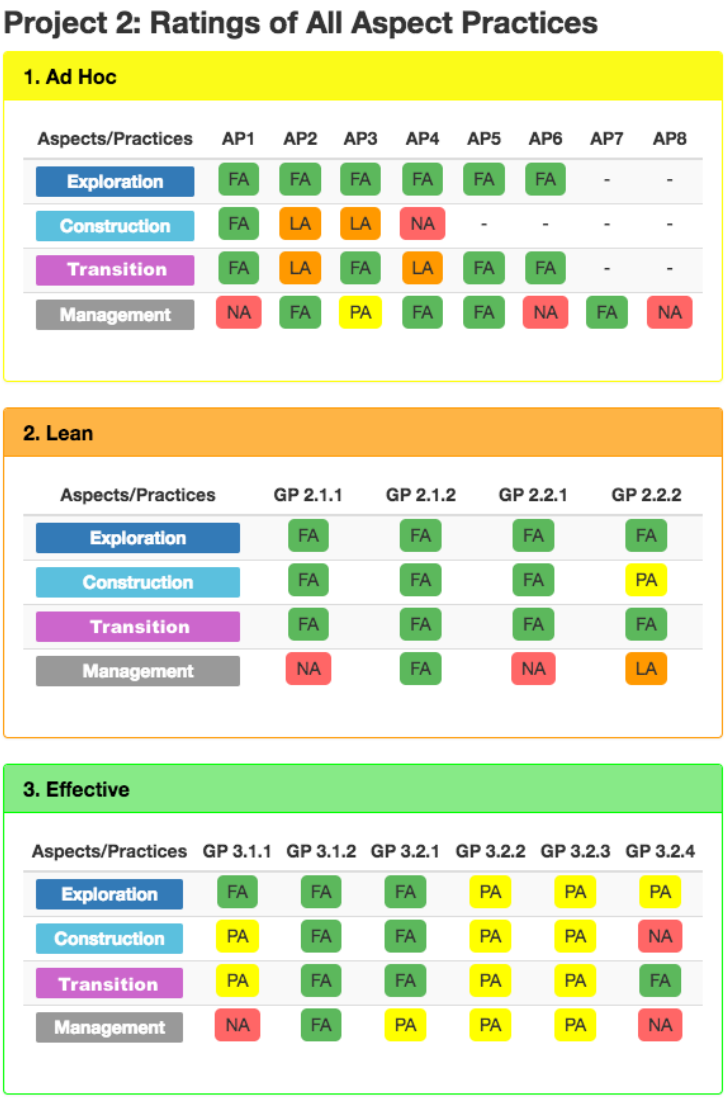


Figure 10: Ratings of All Aspect Practices of Project 2

The assessment report which is generated automatically with the tool, is given at the Appendix F. The aspect based findings revealed through the assessment are given below:

Exploration Aspect - Project 2:

The assessment results showed that the Project 2 achieved Level 2 – Lean in terms of the Exploration Aspect. Out of 16 practices, 13 of them received fully achieved (FA) and remaining 3 achieved partially achieved (PA) rating. The findings, strengths, weaknesses and improvement suggestion for each level are given below:

Exploration Aspect – Level 1 – AdHoc

The mode of operations of the provided solutions to customers are kept as "usage scenarios" and customers select the best fitting scenario for their needs. These "usage scenarios" consists of set of use cases. These sets are discussed with customer to tailor them to their own needs. Then the steps of the use cases are altered or new steps are added or existing ones are moved. These alterations are made in direct meetings with customers. Selected usage stories and tailored use cases are kept in Wrike Tool. The dependencies of the usage stories and use cases are established within the tool.

The product backlog is created and managed through the Wrike Tool. The changes to the items are tracked and necessary adjustments (i.e. re-priotizing) are made to the backlog as changes appear. All the requirement artifacts are visible to both customer and team members through Wrike. Sometimes role based access control feature of the tool is utilized to separate technical items from generic items.

Exploration Aspect – Level 2 – Lean

Requirement artifacts are developed in an iterative and incremental way. Team works in a shared space and communication between team members are supported with daily stand up meetings. Also, the communication feature of Wrike (@mention) is used to communicate anytime anywhere within the team and the customer. The flow of the work is balanced through regular cycle planning gatherings between team members. Requirements are approved informally. The Wrike tool is used for informal decision making with the involvement of the customer.

Exploration Aspect – Level 3 – Effective

Backlog approach is used to keep requirement items in prioritized order. Team members share responsibility of the requirements all together. They own the requirements and manage them collaboratively. Also, the team members work in a unison without a command and control approach however there's still a project manager role which manages teams with command and control style approach. Knowledge is shared between team members with mentoring approach. Measures provided by the Wrike tool are collected however not used.

- Project Manager employs command and control style of approach. To avoid this, trainings should be arranged on Agile Leadership Styles and their benefits.

- There's no agile specific learning approach visible in the organization. To build an Agile centric organizational culture and avoid discrepancies, upper management support should be attained.

- Observation is used instead of interpreting metrics. A strategy about establishing how to interpret the collected measures is needed.

Construction Aspect - Project 2:

In terms of Construction Aspect, Project 2 scored Level 0 – Not Implemented. This was due to the fact that out of 14 practices two received not achieved (NA) rating. Four practices received partially achieved (PA) rating, two received largely achieved (LA) rating, and the remaining six received fully achieved (FA) rating. Below the findings, strengths, weaknesses and improvement suggestions for each level are given:

Construction Aspect – Level 1 – AdHoc

Use cases located in the backlog are elaborated by the team member working on it. Just in time detailing is done. Each developer creates the design about the backlog item he/she is working on and codes it. However, the comprehensibility of the code is reduced because of lack of comments.

- Lack of design discussions between team members is an issue. Designs can be communicated between team members to discuss alternative solutions and approaches.

- There's no coding standard or commenting mechanism to facilitate readability of the solutions.

- Developer level tests are not consistent. Some developers perform automated tests and some are not.

Construction Aspect – Level 2 – Lean

Software is developed in an iterative and incremental way. Backlog is used and frequent demos are made to the customer. Development Team shares the same room and frequent customer visits are made to the development team to discuss changes and for demos. The Wrike tool is again used for distributed communication between team members and customer. The flow of the work is balanced through regular cycle planning gatherings between team members. Retrospectives are held for identification of non-value added activities. However, ceremonies are made with the project manager.

Project management approach and project manager role should be aligned with the Agile approach which is established in team level.

Construction Aspect – Level 3 – Effective

Pair programming is applied but there isn't a specific approach to ensure correctness of the software. GIT tool and committing mechanism used for version and change control. Team members select tasks on their own, the responsibility of the code is shared between members. The parts of the software that requires interaction between each other are developed collaboratively. Team features a champion that shields the development process from the upper project manager. However, there's still command and control approach on the project assignment level. Learning is encouraged within the team. Mentoring and pair programming are used as learning mechanisms.

- TDD approach can be adopted to ensure the correctness of the software.
- Project manager should be trained on Agile Methodologies.
- No organizational learning objectives are in place about agile practices. Organizational learning objectives should be set.
- Construction related measures are not collected. Therefore, a measurement strategy that is in line with organizational and business objectives could be employed.

Transition Aspect - Project 2:

The Transition Aspect of the Project 2 is at Level 2- Lean. Out of 16 practices, three of them rated as partially achieved (PA), and two of them rated as largely achieved (LA). Remaining nine received fully achieved (FA) rating. The level based findings, strengths, weaknesses and improvement suggestions are given below:

Transition Aspect – Level 1 – AdHoc

GIT is used for configuration control. Changes made to the artifacts are made with check-in and check-out mechanisms. Integration is performed automatically with scripts. System can be built with a single command. Deployment is performed continuously and automatically. There're separate environments for development and deployment. Each deployed build is tested by the testers. Regression, integration and acceptance tests are run on the deployed builds by the testers. Acceptance criteria on usage stories are used for the test cases. Internal and external stakeholders can reach to the deployment environment. Build and deployment statuses can be viewed by the customer on the GIT. Also test results, defects found and solutions are published on the Wrike tool for everyone to see, edit and make comments.

- Automated tests can be utilized after integrations to ensure rapid feedback.

Transition Aspect – Level 2 – Lean

Transition aspect activities are performed in multiple iterations. The deployments are planned within backlog planning. Frequent demos are made to the customer. Team works in a shared space and communication between team members are supported with daily stand up meetings. Also, the communication feature of Wrike (@mention) is used to communicate anytime anywhere within the team and the customer. No additional ceremonies are held except demonstrations with customers. No additional meetings are held.

Transition Aspect – Level 3 – Effective

Continuous integration is in the stages of adoption but manual tests are employed still. Check-in and check-out numbers and deployment frequency are tracked. The collected measures are analyzed regularly to track and improve the delivery

frequency to customer. The responsibility of the transition activities is shared between members.

- Automated test suites can be utilized to test the deployments.
- No organizational learning objectives are in place about agile practices for transition.

Management Aspect - Project 2:

The assessment results showed that, Project 2 attained Level 0 – Not Implemented at the Management Aspect. Out of 18 practices seven practices received not achieved (NA) rating. There are four practices rated as partially achieved (PA), six practices rated as fully achieved (FA), and one practice rated as largely achieved (LA). The findings, strengths, weaknesses and improvement suggestion for each level are given below:

Management Aspect – Level 1 – AdHoc

The projects do not include separate feasibility studies conducted or vision statements. Scope is defined with the selected usage stories. The project team is formed according to the experience and knowledge. Customers are aware of the fact that agile is applied through development. However, there's an inconsistency between project manager and the team. Physical workspace is appropriate for agile. Shared space, quiet, and solitary rooms are available for all team members. The development plan is prepared and maintained iteratively. Daily activities are coordinated through daily stand up meetings. Estimations are done via expert judgment. The progress of the team is monitored by the project manager with the Wrike tool. Effort, schedule and cost are monitored and updated through tool's dashboards. Project risks are not tracked.

- Initiation strategies including feasibility studies could be performed individually for each project.
- Project manager should be trained on Agile Methodologies.
- A proved estimation methodology could be adopted and historical database could be set for future estimations.
- A risk management strategy that is aligned with organizational objectives could be developed and used.

Management Aspect – Level 2 – Lean

Management related work products are not developed in an iterative and incremental way. Project Manager and team member communicates effectively through weekly meetings. Customer is involved in the meetings. Project management related decisions are not taken collaboratively. Informal procedures are applied to handle management decisions. Project management plans and estimations could be

developed iteratively and incrementally as more information obtained through development cycle.

- Project manager has absolute responsibility over management decisions but team members should be given opportunity to have their say for management related work.

Management Aspect – Level 3 – Effective

Agile methods are not incorporated to management work. Collaboration between project manager and team champion is visible. There's an underlying estimation collaboration between team members and champion. However, the team is not directly involved in planning and estimation processes. Project Manager still employs command and control style of management on the team. However, teams have leaders that enables people to work without command and control mechanism. Learning is encouraged within the team. Mentoring and pair programming are used as learning mechanisms. Management aspect's activities are not followed through measures.

- Effort estimation could be based on historical data or function point estimation.
- Team should be involved in planning and estimation processes.
- Project manager should be trained on Agile Methodologies.
- There is not an organizational learning objective in place about agile practices for management. To be aligned with Agile as a whole, an organizational learning strategy and plan could be created.
- Managers should track and monitor their work and discuss the findings with teams and each other.

5.1.3.3 Case Study 3

The assessment for the Case Study 3 is performed through a total of 2,5-hour interview session with the game leader and a software developer. Each person is interviewed by one of the assessors and whole interview is completed within total of 10 person-hours. The details of the Case Study 3 are given in the table below:

Table 23: Information about Case Study 3

Organization ID:	Organization BV
Size of the Organization:	Small SME (25-100 Employees)
Agile Experience:	2 Years
Organization	Organization BV is an indie game development studio that develops cross platform game applications. In total the studio

Bio:	has released 18 mobile games in different genres for Android and iOS mobile platforms. The Agile is the agreed software development method for the studio since the beginning and a Kanban based customized method is used for the projects.
Project ID:	Project 3
Project Bio:	The project assessed in the scope of the study was a new development project that involves development of a third person action adventure game. The game is based on the neural network AI technologies and has the ability to adapt the main scenario to player's gaming style. The project was developed with the Unity Game Engine.
Project Team Bio:	The project was carried out by a team that consists of 9 members. The distribution of the roles is given below: • 1 Game Leader • 1 3D Storyboard Artist • 2 Graphic Designers • 3 Software Developers • 1 Voice Engineer

Assessment Results:

We assessed the Organization BV over the Project 3 which new development project that involves development of a third person action adventure game. The achieved agility levels for the project can be seen on Figure 11 the below:

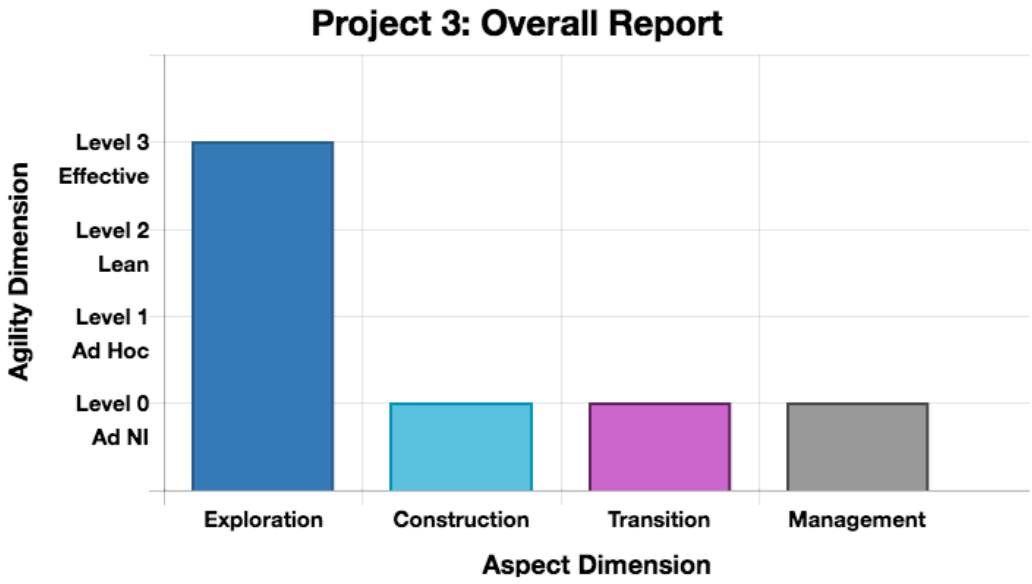


Figure 11: Achieved Agility Levels of Project 3

The results show that Exploration Aspect is at the Level 3: Effective which means Agile engineering methods/practices are internalized, tools are integrated to aspects, and learning and improvement are in place [84]. Construction, Transition, and Management Aspects are at Level: 0 Not Implemented which means that Aspect Practices are not achieved or partially achieved [84]. The ratings of the all aspect practices are given at the Figure 12 below:

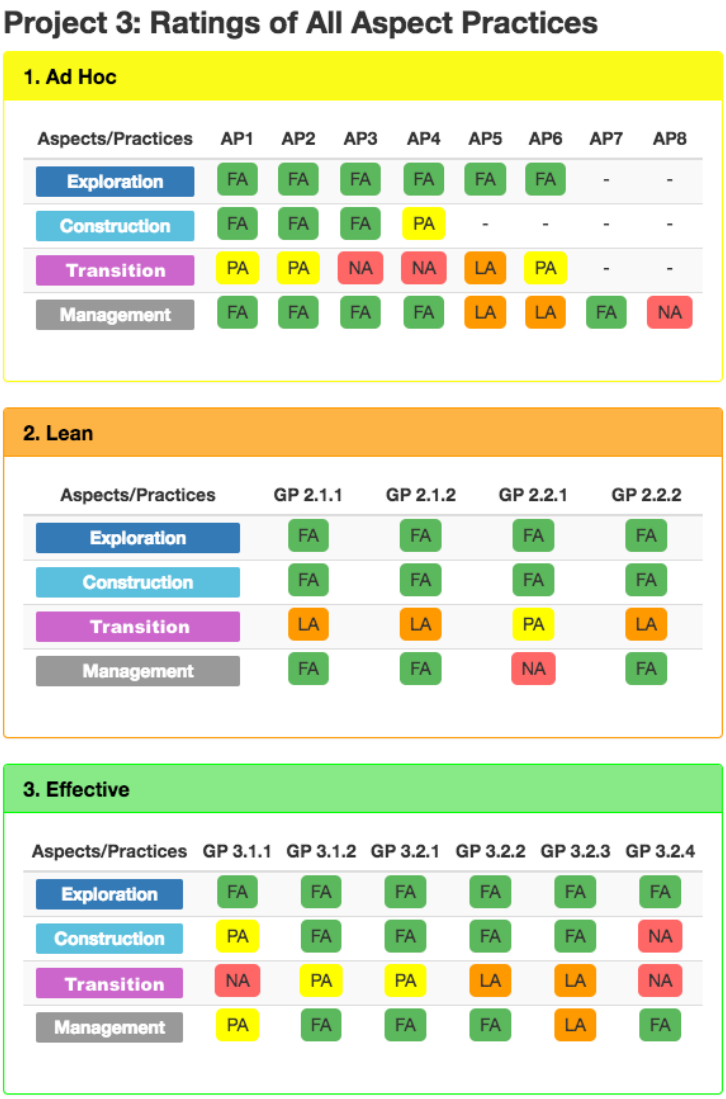


Figure 12: Ratings of All Aspect Practices of Project 3

The assessment report which is generated automatically with the tool, is given at the Appendix F. The aspect based findings revealed through the assessment are given below:

Exploration Aspect - Project 3:

In terms of Exploration Aspect, Project 3 attained Level 3 – Effective with all 16 practices rated as fully achieved (FA). The major findings, strengths, weaknesses and improvement suggestions about the Exploration Aspect are given below:

Exploration Aspect – Level 1 – AdHoc

The Project 3 is a mobile game application project and there is an idea board that is used for the team members to write their game ideas. At regular intervals, these

written ideas are voted and the ones that made the top 3 are picked for the development. Then the owner of each idea writes the high-level requirements for each game. After the owner writes the high-level requirements, Game Design Documents including the story, game item, objectives and dependencies are prepared. Then, rough sketches for the screens are designed. Each sketch is designed addressing specific requirements (taken as side notes with the screens) and exported and stored on the Mural tool. Mural enables the connection between high level requirements and sketches. The conflicts related to the requirements are resolved with team members discussing with each other on white board. The Mural tool is used to specify and keep the dependencies between requirement artifacts. High level requirements, screen designs, graphic designs and notes are related to each other. A Kanban board is kept to manage and prioritize the work items. Changes are discussed collaboratively and re-prioritizing is done when a change occurs.

Exploration Aspect – Level 2 – Lean

Requirement artifacts are developed in an iterative and incremental way and made visible to everyone by the Mural tool. Screen designs enable visual demonstration from early phases and detection of misunderstandings. Team shares the same room and daily stand-up meetings (or as the team says grab-a-coffee) meetings are held to communicate and discuss daily activities and problems. Work is balanced through the Kanban board. Team commits to limiting WIP. Team plans cycles by choosing the work items that will be developed in that cycle. The team has a policy that they're calling just-make-it-happen. The policy states that no ceremony should be held unless it's necessary. Also, the team members state that they're applying RUP rules to the documents that is: "If document does not increase your understanding of the product then it's unnecessary".

Exploration Aspect – Level 3 – Effective

Kanban practices such as visualizing the work, limiting WIP, managing flow, explicit process and feedback loops are applied. Team collaborates as a whole and it's self-organizing. The team members are sharing the responsibility of the game however they're organized according to their interests and expertize. The Game Leader leads the team with agile leadership styles and focuses on guiding people to do their works. Mistakes and problems are discussed without blaming and each mistake is embraced as a lesson learned. The Mural is used for team members pointing the new technology that they should learn and apply. Also, teams are constantly attending to educational events and conferences on the subjects such as UX Design and Agile Methodologies. The number of screens is used as a measure to track the requirements. Then the team categorizes the screens according to a complexity scheme they have developed from previous experiences. This metric is collected and analyzed for size and effort estimates.

Construction Aspect – Project 3

The Project 3, scored Level 0 – Not Implemented in the Construction Aspect. Out of 14 practices two of them rated as partially achieved (PA) and remaining ones are all

rated as fully achieved (FA). The major findings, strengths, weaknesses and improvement suggestions about the Construction Aspect are given below:

Construction Aspect – Level 1 – AdHoc

In the exploration activities, the requirements are visualized with rough sketches. In the construction, each sketch is turned into detailed screens with specific elements and solutions (taken as side notes with the screens) then exported and stored on the Mural tool. Mural enables the connection between high level requirements, sketches and detailed screens. User Interface Prototyping is used as the default design exploration technique. Also, some teams are trying to adapt UX design first approach. Software is developed with Unity Game Engine. This enables the team to develop platform independent games. Software is tested manually on the user interfaces.

- There's no coding standard.
- Peer reviews, pair programming and automated unit tests can be applied to ensure the correctness of the software.

Construction Aspect – Level 2 – Lean

Software is developed in iterations. The games are divided into functional cohesive parts and each part is developed iteratively. Team shares the same room and daily stand-up meetings (or as the team says grab-a-coffee) meetings are held to communicate and discuss daily activities and problems. Kanban Methodology is used effectively to limit WIP. Ceremony is internally minimized and document writing criteria is in place.

Construction Aspect – Level 3 – Effective

Prototyping is used for determining strengths and weaknesses of a suggested solution. MS Visual Studio is used as an integrated development environment and GIT is used for configuration management. The team collaborates as a whole and it's self-organizing. Team members are sharing the responsibility of the game however they're organized according to their interests and expertize such as 3D design and voice engineering. The Game Leader leads the team with agile leadership styles and focuses on guiding people to do their works. Mistakes and problems are discussed without blaming and each mistake is embraced as a lesson learned.

- Correction of the solution is not verified. Pair programming and coding standards can be applied to increase quality of the code and TDD and unit tests can be used to ensure the correctness.
- No measures are taken for the construction aspect activities. Code quality metrics such as defect density and check-in and check-out numbers can be collected and analyzed.

Transition Aspect – Project 3

The assessment results revealed that Project 3 attained Level 0 – Not Implemented in the Transition Aspect. Out of 16 practices Project 3 attained not achieved (NA) rating for four practices and remaining 12 practices evenly rated with partially achieved (PA) and largely achieved (LA) ratings. The major findings, strengths, weaknesses and improvement suggestions about the Transition Aspect are given below:

Transition Aspect – Level 1 – AdHoc

There's only one environment for development and test. Code is under configuration control and changes are stated with comments when the code is committed. The code is integrated through check-in and check-out mechanisms. Deployment is performed however it's nor automatic nor continuous. There's no explicit testing mechanism to test integrated solution. Tests are done manually via graphical user interface. Transition process is visible to team members. Tutorials are prepared at the transition stage.

- There isn't an explicit testing approach. Changes to development items are not linked to other related artifacts.
- Integration frequency is low (one integration in 2 days). The frequency of integration can be increased. Automated integration and test mechanisms can be applied.
- A testing approach can be developed and adopted to ensure correctness
- Lack of maintenance documentation causes latencies during update cycles. Other documents for the maintenance of the software can be produced.

Transition Aspect – Level 2 – Lean

Transition activities are performed iteratively and incrementally. But, the iteration length is inconsistent. The team has a policy that they're calling just-make-it-happen. The policy states that no ceremony should be held unless it's necessary. Only documents created in this aspect are tutorials and integration comments.

- Iteration length for the transition activities should be established.
- Collaboration is not fully obtained within the team. Especially, team members in testing and development could collaborate through the transition.
- Limiting WIP principle seems to be not implemented to the transition aspect activities. The integration iteration lengths are inconsistent some are 4 weeks long some are a week long.
- Criteria for maintenance documents should be established to ease the maintenance.

Transition Aspect – Level 3 – Effective

Agile Practices are not internalized for this aspect. GIT tool is used for deployment. Not all team members share the responsibility for deployment and integration. Only members who integrated the code are regarded responsible. No measures are collected during transition phase.

- Continuous integration, and Integration and acceptance tests should be performed.
- Deployment is not automated.
- Responsibility of the deployment can be shared between members with consistent integration iterations that enables everyone to integrate their code.
- Learning objectives on continuous integrations and integration and acceptance testing could be set.
- Defect analyzes and integration frequency could be collected and analyzed.

Management Aspect – Project 3

In the final aspect: Management, Project 3 again attained Level 0 – Not Implemented. Out of 18 practices, three practices received not achieved (NA) rating, one practice received partially achieved (PA) rating, three practices achieved largely achieved (LA) rating and remaining 12 practices received fully achieved (FA) rating. The level based findings, strengths, weaknesses and improvement suggestions are given below:

Management Aspect – Level 1 – AdHoc

The feasibility study for the games are conducted based on the trend analysis. The popular game genres are examined and brainstorming is done to create ideas that conforms to the results of the trend analysis. Then ideas are recorded (as Game Design Documents) to the Mural tool and ranked based on voting. Teams are formed based on the expertize areas. Teams include storyboard designers, graphic designers, software developers and sound artists. Team and upper management are aligned with the Agile values and principles. Management embraces Agile and supports the practices. Physical workspace has separate parts where team members can work together or work in isolation. Offices includes whiteboards, games and other elements to facilitate collaboration and thinking aloud. Release and sprint plans are prepared. Estimates are done based on the detailed screen since the time to develop similar screens are known. Daily stand-up meetings are held to monitor and improve the progress. Retrospective meetings are held to review the projects and discuss the lessons learned. Estimates are done based on the detailed screen since the time to develop similar screens are known. There's a directory to keep actual past development efforts and times. The Game Leader monitors the progress of the team based on the screens completed and effort spent on them. The leader discusses these parameters regularly with the upper management. Risk mitigation is not done.

- The Plans are not updated regularly through the changing conditions of the project. The plans could be updated regularly with the changing requirements and conditions. Project Manager and Teams should steer the plans together.

- The actual values entered by the team members includes noise and causes deviations in estimates. A strategy for entering actual effort data could be generated to reduce the noise.

- A risk tracking, analysis and mitigation strategy can be employed to manage and mitigate the risks.

Management Aspect – Level 2 – Lean

The plans, progress reports and project tracking activities are performed in multiple iterations. Daily stand-up meetings (or as the team says grab-a-coffee) meetings are held between Game Leader and team members to communicate and discuss daily activities and problems. Also, weekly progress meetings and retrospective meetings are held between Game Leaders and upper management. Plans and estimations are not updated continuously during the course of the project. Retrospective meetings enable non-value added activities to be eliminated from the process. Dashboard prints and automated reports are used as documentation to monitor and track the project progress.

- Plans could be updated regularly with the changing requirements and conditions. Project Manager and Teams should steer the plans together.

Management Aspect – Level 3 – Effective

Estimations are done with collaboration between Game Leader and Team Members. Mural and plan.io are used as management tools. Estimation and planning are collaborative activities. Team members make their own estimates for the jobs they've assigned. The Game Leader leads the team with agile leadership styles and focuses on guiding people to do their works. Mistakes and problems are discussed without blaming and each mistake is embraced as a lesson learned. The Game Leaders share their knowledge and experiences through retrospective meetings to learn from each other. Management aspect activities are tracked and monitored through measures such as plan and estimate accuracies.

- Estimations are not improved continuously during the course of the project.

- The Management Aspect based learning objectives can be set.

Findings on Research Questions

Findings on RQ1: To what extent is the tool sufficient to meet the expected criteria?

Q1: Please rate the tool with the evaluation questionnaire provided.

The ratings given by assessors and the lead assessor are given in the table below. Considering the ratings given, it can be said that the AssessAgility has fully

complied with the 8 out of 9 expected criteria: Coverage, Availability, Guidance Capability, Assessment Recording, Automated Reporting, Comparability, and Extensibility. Only criterion that could not be fully complied was Different Scopes.

The detailed comments given by the lead assessor and assessors are given below.

Table 24: Evaluation of the Expected Criteria

Criteria	Lead Assessor	Assessor 1	Assessor 2
Coverage	**FA**	**FA**	**FA**
Availability	**Web-Based**	**Web-Based**	**Web-Based**
Guidance Capability	**FA**	**FA**	**FA**
Assessment Recording	**Yes**	**Yes**	**Yes**
Automated Reporting	**Yes**	**Yes**	**Yes**
Comparability	**Yes**	**Yes**	**Yes**
Different Scopes	LA	LA	LA
Extensibility	**Yes**	**Yes**	**Yes**

Coverage:

Lead Assessor: "The design of the reference model is based on the Agile Manifesto itself and not on a set of Agile practices. Therefore, by using the reference model, the tool provides full coverage on the Agile Manifesto."

Assessor 1: "Since the tool is employing the AgilityMod, it fully covers the 12 Agile principles. "

Assessor 2: "The aspects and practices in the model enables tool to cover all the values and principles in the manifesto."

Availability:

Lead Assessor: "Tool is available on the cloud platform and via the provided link it's easy to connect and start using the tool."

Assessor 1: "Tool is available through the Web."

Assessor 2: "Tool is reachable online with an internet explorer."

Guidance Capability:

Lead Assessor: "The tool has informative notes on every page that includes simple and easy to understand instructions. These notes also give information about the steps of the assessment process. The assessment section includes notes about the reference model and provides elaborative warnings about the practices. Also, each primary action has a warning pop-up that informs the user about his/her actions. The automated reports have more than enough fields to interpret the results of the assessment and floating action button is a nice touch that guides the user throughout the process."

Assessor 1: "The tool guides the assessment process seamlessly. It includes guidance facilities for conducting the assessments. It also has nice to have features such as the informative fields that gives information about the status of the assignments, instructions on each page and distinctively colored warnings."

Assessor 2: "It is extremely easy to perform assessments with the tool. It guides the user about what to do next and the progress left. The guidance features that helped me most are: the warning pop-ups, progress bars and warning areas that attract notice to the things that requires attention."

Assessment Recording:

Lead Assessor: "The tool records the assessments for further analyzes and reporting. It displays all the recorded assessments in lists and provides status info with color codes."

Assessor 1: "The tool records assessments and lets user to pause and then continue to conduct the assessments."

Assessor 2: "Tool keeps both the completed and uncomplete assessments and provide a list to view them as necessary."

Automated Reporting:

Lead Assessor: "The tool generates automatic reports for specific needs of the assessed party. It provides short reports for high level management, detailed reports for process improvement bodies, comparison reports to compare different projects and combination reports for assessments that are conducted concurrently by different assessors."

Assessor 1: "Tool provides different types of reports for different types of stakeholders. All the reports are generated automatically."

Assessor 2: "Tool enables generation of reports automatically."

Comparability:

Lead Assessor: "Tool enables to make comparison between assessments on achieved agility levels. The comparison results are provided as comparison reports."

Assessor 1: "The tool automatically creates comparison reports for the assessments that needs to be compared."

Assessor 2: "The comparison report facility enables comparison of the assessment results."

Different Scopes:

Lead Assessor: "Tool provides assessment of project and the related team however it does not have an explicit feature that enables conducting assessments on organizational level. However, this drawback is caused by the structure of the reference model."

Assessor 1: "Due to the reference model tool only allows conducting on assessments for the scopes: team and project."

Assessor 2: "Tool only enables assessors to conduct assessments on project level. However, it's possible to deduct results for the team that is responsible for the project. The tool does not provide assessments for the organizational scope."

Extensibility:

Lead Assessor: "The independent assessment feature of the tool lets modification of the assessments for different contexts."

Assessor 1: "Tool enables flexibility on the assessment process with the independent assessment feature."

Assessor 2: "Tool lets users to change the assessment scope and process."

Findings on RQ2: To what extent is the tool able to automate and guide the assessment process?

Q2: Does the tool cover the assessment process as a whole? If not, please describe the missing parts of the process.

Lead Assessor: "The tool facilitates all the phases described in the exemplar assessment process. It covers Planning Phase with the features provided in Projects, Teams and Assignments pages. It facilitates Data Collection Phase with Perform Assessment page. It facilitates Validation and Agility Level Determination Phase with Analyze feature provided to lead Assessor and facilitates Reporting Phase with automated reporting function. "

Assessor 1: "Tool covers all the assessment process regarding the assessors (Data Collection Phase) as a whole."

Assessor 2: "Tool provides facilities for the process elements regarding the assessors."

Q3: Does the tool guide the assessment process as a whole? If you required additional guidance about the assessment process, please describe the parts where you needed guidance?

Lead Assessor: "With the instructions, warnings and clearly defined and sorted pages, the tool provides guidance from Planning Phase through the Reporting Phase."

Assessor 1: "It is easy to perform assessments with the tool as the Perform Assessment page is designed compatible with the steps of the process."

Assessor 2: "Tool enables keep tracking of the process and provides sufficient guidance with the instructions, warnings, pop-ups and pages aligned with the process."

Findings on RQ3: How efficient is the tool for the assessment process?

Q4: Please rate the usability of the tool with the questionnaire provided.

The ratings given by each team member for each area of the questionnaire is given below:

Lead Assessor:

Table 25: Lead Assessor USE Questionnaire Ratings

Usefulness: 53/56	**Ease of Use:** 75/77
Ease of Learning: 28/28	**Satisfaction:** 49/49

Assessor 1:

Table 26: Assessor 1 USE Questionnaire Ratings

Usefulness: 54/56	**Ease of Use:** 72/77
Ease of Learning: 27/28	**Satisfaction:** 49/49

Assessor 2:

Table 27: Assessor 2 USE Questionnaire Ratings

Usefulness: 55/56	**Ease of Use:** 77/77
Ease of Learning: 28/28	**Satisfaction:** 49/49

The questions in the USE questionnaire are constructed as seven-point Likert rating scale, a rating of seven means strong agreement and a rating of zero means strong disagreement. The ratings suggest that assessment team strongly agrees to the positive statements in the questionnaire therefore according to the USE Questionnaire, AssessAgility's usability is high in terms of the questionnaire's three dimensions: Usefulness, Satisfaction and Ease of Use.

Q5: Would you prefer performing agility assessments with the tool or without the tool? Why?

The responses taken from the assessment team members are given below:

Lead Assessor: "I definitely prefer performing agility assessments with the tool, because:

- The effort required to organize and perform assessments is much more reduced according to the manual assessments
- Reports are generated automatically
- Tool enables working as distributed teams and provides coordination for distributed teams
- Tool eliminates the need for looking at documentation
- Assessments can be done everywhere and anytime"

Assessor 1: "I prefer the tool because it provides:

- Automatic reports
- Guidance for assessments
- Keeps track of the structure of the assessment and easy navigation through the aspects
- Fields for entering mandatory and optional findings and notes
- Useful user interface and easy to understand flow
- Remote working
- Useful features such as pausing the assessment, warnings and informative notes"

Assessor 2: "I prefer performing assessments with the tool because:

- I was able perform assessments without looking at the reference model,

- I didn't get confused because it automated the process and provided instructions,
- I've been able to easily reach it every time I try,
- It kept track of the progress and I was able to manage the time,
- It gave me warnings about the due dates of the assessments assigned to me,
- It provided me reports while still performing the assessment,
- Entry of the data was much simpler than the manual assessment."

Q6: Does the tool reduce the effort spent for performing assessments? Could you please describe the difference?

For this question, we requested from the assessment team members to perform a manual assessment for a pilot case and we wanted them to record the effort that they have spent for the manual assessment and compare it to the average effort that they have spent for the assessments performed for each case in the multiple case study. The reasons for the difference and the rate of the effort difference are presented below:

Lead Assessor 1: "Tool definitely reduced the effort I've spent for assessments because it enabled me to manage the process with just simple steps. I was able to make assignment just in minutes and get reports with just a click of a button. Automatic reporting facility definitely reduced the time I've spent during manual assessment and the difference is around 50%."

Assessor 1: "I didn't have to enter the details for the assessment and I didn't have to keep track of the assessment process myself, therefore, the tool eliminated the time I've spent for the redundant work. If I compare the time I've spent for the manual assessment that I've conducted for the training with the time I've spent for the automated assessment, there's a clear difference of 40%."

Assessor 2: "Since I didn't have to look at the documentation the tool reduced 20%-30% of my time for performing an assessment."

The comparison of the effort spent based on the phases that each assessment team member could be find on the Figure 13 below:

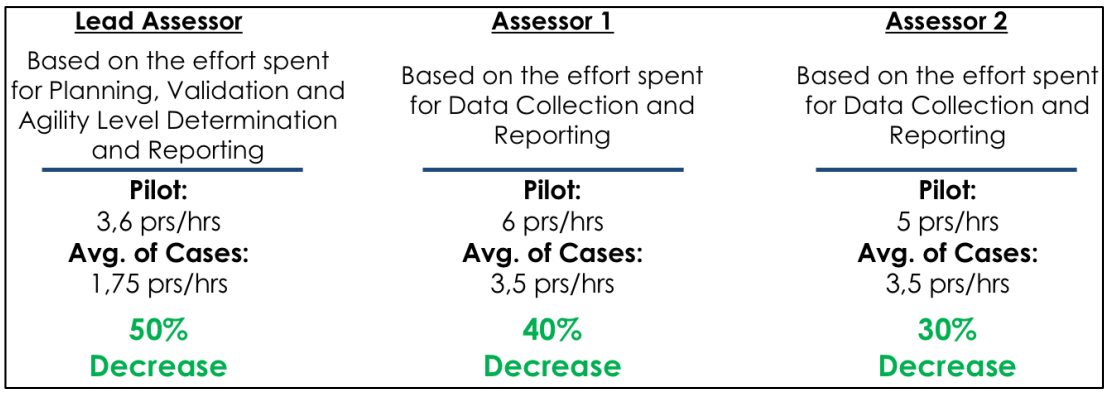


Figure 13: Effort Comparison Diagram

Findings on RQ4: What are the strengths and weaknesses of the tool?

Q7: Could you please describe the strengths and weaknesses of the tool?

The answers from all three team members are given below:

Strengths:

- Ease of management of assessments
- Ease of assessment performance
- Ease of use and understanding
- Clear, responsive, and useful user interface
- Guidance for the assessment process
- Automated reports
- Support for parallel and distributed assessments
- Easy data and evidence collection

Weaknesses:

- Tool requires an instant messaging facility which could ease the coordination between distributed team members
- Notification feature to let users about the assignments and their status

5.1.3.4 Validity Threads

Due to the fact that multiple case study approach employs qualitative data, it's prone to validity threads. According to Yin [96]; the quality of case studies can be evaluated by four tests: construct validity, internal validity, external validity and reliability. Below, the techniques employed to address each test is given with details.

Construct validity concerns with how well an experiment aligns to its claims. For ensuring construct validity, we gathered evidences from multiple sources: documentation, interviews and participant observation. These sources are three of the

total of six sources that Yin [96] suggested. Furthermore, the evidence is stored with the assessment reports in the database to ensure the construct validity by storing and then we gave access to experts to get feedback about the validity of the case study.

Internal validity concerns with achievement of casual conclusion by minimizing the systematic error or bias. In order to ensure internal validity, we employed explanation building technique suggested by Yin [96]. We discussed the assessment results within reviews held with the attendance of every team member in the assessed projects. We present evidences and explain the relationship between each evidence and the obtained practice rating. Each rating and evidence is discussed with team members to avoid any bias in the results.

External validity concerns with generalization of results to other contexts. In order to ensure external validity, we performed assessments with three different organizations working in three different business domains, and developing three different types of software. (See Tables 44, 45, and 46 for the information on cases.) Furthermore, for each assessment two assessors gathered separate evidences and these evidences are reviewed by lead assessor. Therefore, the tool is used in three different settings with three different people and we believe that this will increase the generalizability.

Reliability concerns with the repeatability of the study by eliminating errors and biases. To ensure reliability we employed multiple case study approach and we replicated the results by performing assessments on same project (or same setting) with two different assessors.

CHAPTER 6

CONCLUSION

This chapter includes a summary of this thesis study and contributions achieved by the proposed agility assessment tool and exemplar assessment process. Lastly, the suggestions and direction for the future work are included.

6.1 Summary of the Study

Agile Software Development Methodologies have gained massive popularity since their emergence. The software community has been supportive and eager towards the Agile, because these methodologies are proved to be flexible and convenient. However, random adoption of Agile practices and misinterpretation of values and principles, preclude adopters to gain full benefit from Agile. To solve this problem, researchers and practitioners have developed agility assessment tools to help adopters to determine their agility gaps and seek out improvement suggestions for increased Agility. The suggested methods fall under three distinct categories: agility assessment models, checklists and surveys. The checklists and surveys are unable to provide a comprehensive solution due to the fact that they are developed for specific Agile methods and their reliance on predefined sets of Agile practices. Models provide a more comprehensive solution however the time and effort required to apprehend and implement the models are very high. To reduce the effort and time required to apply the models, agility assessment tools were provided for the use of adopters and researchers.

To find out these tools, we performed a systematic literature review on scientific databases and we initially identified 42 agility assessment tools. After, elimination of 20 tools that do not have the ability to automate and guide the assessment process. And 12 unavailable ones, our literature review revealed 14 templates and software programs that have the ability to reduce the time and effort required to perform agility assessments.

Then, to identify the features of these tools and set requirements for a better agility assessment tool, we performed a multiple case study that includes 14 agility assessments made with 14 accompanying tools. Within the case study, we also evaluated the tools according to the identified features or as we call them: expected criteria. In the light of the features identified and as a result of the case study, we defined nine expected criteria namely: Coverage, Availability, Guidance Capability, Assessment Recording, Automated Reporting, Comparability, Different Modes of

Usage, Different Scopes and Extensibility. The conclusion arrived from evaluation of the tools against these expected criteria is that, none of the tools were able to fully comply with the nine expected criteria.

Furthermore, our case study, revealed that these tools are mainly developed for assessing agility in limited conditions and contexts such as the beginning of an agile adoption process, certain implementations such as a Kanban implementation, on team, organization and project basis. Additionally, the majority of the tools were still relying on a set of agile practices to indicate the level of agility. While these practices are crucial for specific implementations of agile methods, the mere absence or presence of these practices is not sufficient to indicate the success of the adopted agile method. In addition to that, majority of the tools do not provide agility levels or the possible improvement areas towards agility. Finally, we observed that the tools are mostly focused on conducting the assessment but lacking the support for other important parts of the assessment process such as planning and data validation. Therefore, in addition to fully satisfying our nine criteria and having built upon structured agility assessment models, the tools are expected to have features that facilitate and automate the whole assessment process including planning, conducting, and reporting the agility assessments to reduce the time and effort spent for the assessments.

To address the findings and problems listed above, we proposed a web based generic agility assessment tool, that has the ability to guide and automate the assessment process as whole and is applicable to every Agile methodology including the custom ones. The software requirements of the tool are specified according to the nine expected criteria we have defined as a result of the case study for full compliance.

To make the tool generic, we based it on the structured and most recent Agility Assessment Reference Model, AgilityMOD. Since AgilityMOD is built upon the principles and values in the Agile Manifesto, it gave the tool the ability to be applicable to any adoption without reliance on a specific Agile methodology. In addition, it enabled the tool to yield comprehensive assessment results, that include agility gaps and improvements suggestions to address those gaps.

To provide full guidance and automation abilities, we first defined an exemplar agility assessment process for the reference model and specified the initial mode of operation of the tool in compliance with the process. On the top of that, we included guidance notes and instructions to the tool that helps its users to follow the process swiftly. With this inherent design elements, the tool attained the ability to fully guide and automate the assessment process intrinsically. After the specification of the requirements and design of the tool, we developed it as a cloud based web application in four two-week long time-boxed iterations.

When the implementation completed, we designed a multiple case study to validate our proposal. The multiple case study included three separate agility assessments on three different projects from three distinct organizations that develop software solutions on different business domains. Assessments were performed according to the exemplar process model defined with an assessment team which is independent from organizations and the researchers involved in this thesis. The assessment team

consisted of a lead assessor and two assessors. In the scope of the multiple case study we evaluated the tool's compliance to the nine expected criteria we have defined and its ability to provide full guidance and automation. In addition to that, we requested from the assessment team to test its efficiency and report the strengths and weaknesses they identified while performing the assessments.

In summary, the results of the multiple case study showed that according to the assessment team members, the tool has the ability to guide and automate the assessment process as a whole. In terms of the expected criteria, the tool proved to be fully achieved the Coverage, Availability, Guidance Capability, Assessment Recording, Automated Reporting, Comparability, Different Modes of Usage and Extensibility criteria and largely achieved the Different Scopes criteria due to its lack of support for performing assessments on organizational level. The proposed tool's performance according to the expected criteria and its comparison against the other tools that have been evaluated in the scope of the study can be seen on Table 51 on the next page.

Table 28: Comparison of the Agility Assessment Tools

#	Tool/Quality Criteria	Coverage	Availability	Guidance Capability	Assessment Recording	Automated Reporting	Comparability	Different Modes of Usage	Different Scopes	Extensibility
1	Agile Assessment	**FA**	Not Web-Based	PA	**Yes**	**Yes**	No	NA	NA	No
2	Agile Enterprise Survey	LA	**Web-Based**	PA	No	No	No	NA	NA	No
3	Agile Health Dashboard	LA	Not Web-Based	**FA**	**Yes**	**Yes**	**Yes**	NA	NA	**Yes**
4	Agile Journey Index	**FA**	Not Web-Based	**FA**	**Yes**	**Yes**	**Yes**	NA	NA	No
5	Agile Process Assessment Tool	LA	Not Web-Based	PA	**Yes**	**Yes**	No	NA	NA	No
6	Agile Self Assessment	**FA**	**Web-Based**	LA	**Yes**	**Yes**	No	LA	NA	No
7	Agility Questionnaire	LA	Not Web-Based	LA	**Yes**	**Yes**	No	NA	**FA**	**Yes**
8	Comparative Agility	**FA**	**Web-Based**	**FA**	**Yes**	**Yes**	**Yes**	NA	NA	**Yes**
9	Depth of Kanban	LA	Not Web-Based	PA	N/A	No	No	NA	NA	**Yes**
10	Enterprise Agility Maturity Matrix	**FA**	Not Web-Based	LA	**Yes**	**Yes**	No	NA	LA	No
11	GSPA	**FA**	Not Web-Based	LA	**Yes**	**Yes**	No	LA	LA	**Yes**
12	IBM DevOps Practices Self Assessment	LA	**Web-Based**	LA	**Yes**	**Yes**	No	NA	NA	No
13	Open Assessments	**FA**	**Web-Based**	**FA**	No	**Yes**	No	NA	NA	No
14	TeamMetrics	NA	**Web-Based**	**FA**	**Yes**	**Yes**	**Yes**	**FA**	NA	No
15	AssessAgility	**FA**	**Web-Based**	**FA**	**Yes**	**Yes**	**Yes**	**FA**	LA	**Yes**

According to the assessment team members, the tool increased their efficiency with an average of 30% against the efficiency rate obtained in the assessments that assessment team members performed manually and it proved its usability according to the USE Questionnaire by scoring average of 206 out of 210 points in total. The overall strengths listed by the assessment team members are given below:

- Ease of management of assessments
- Ease of assessment performance
- Ease of use and understanding
- Clear, responsive, and useful user interface
- Guidance for the assessment process
- Automated reports
- Support for parallel and distributed assessments
- Easy data and evidence collection

And overall weaknesses listed by the assessment team members are given below:

- Tool requires an instant messaging facility which could ease the coordination between distributed team members
- Notification feature to let users about the assignments and their status

All in all, the multiple case study concluded that AssessAgility is able to fully guide and automate the assessment process and reduce the time and effort spent for the agility assessments.

6.2 Contributions Achieved by the Study

The contributions achieved by this thesis study can be categorized under three distinct categories which are given below as separate sub sections:

The Contributions Achieved by the Literature Review

The literature review of this study, presents a comprehensive list of agility assessment tools that are available for assessing agility. The multiple case study conducted within the literature review, identifies the prominent features and shortcomings of these tools and defines the expected criteria for agility assessment tools to have. By using these criteria, the multiple case study also presented a thorough evaluation and comparison of the agility assessment tools that are available on the literature.

The Contributions Achieved by the Exemplar Assessment Process

The exemplar assessment process that is defined within this study, includes the characteristics of the assessment teams, roles and responsibilities, phases and steps required to conduct agility assessments with the reference model AgilityMod. In these terms, the process extends the AgilityMod: Agility Assessment Reference

Model by improving its applicability. Furthermore, the exemplar assessment process increases the uniformity of the results of the assessments conducted with the reference model by providing comprehensive guidance for researchers, organizations and assessors to perform assessments.

The Contributions Achieved by the AssessAgility

The major contribution of this study, is the web based generic agility assessment tool: AssessAgility. The tool is built upon the prominent features of the currently available agility assessment tools and is designed to overcome the shortcomings of the currently available tools such as reliance on a specific set of practices and not providing support for the whole assessment process. Furthermore, the tool enables utilization of the most current and structured agility assessment reference model AgilityMod and is usable on any specific Agile methodology. Finally, the tool provides full guidance and automation for the agility assessment process that increases the efficiency of the assessment process.

6.3 Future Work

In terms of future work, the suggestions given below are regarded:

- Addition of an online messaging feature to the tool for ease of communication and coordination between distributed team members.

- Addition of a notification infrastructure to the tool that has the ability to instantly inform the users of the tool about the events generated in the tool such as assignments made, assessments complete and approaching due dates.

- Addition of a calendar management page for lead assessors to better track the assignments they made for the assessors.

- Development of an approach for the AgilityMOD and accompanying feature for the tool to perform organization level assessments.

- Development of a benchmark data release approach for the data captured with the tool.

- Development of a self-assessment approach that provides a set of predefined questions and exemplar answers and examples that are compatible with the comprehensive structure of the AgilityMOD, and integration of this approach to the tool's self-assessment feature.

- Addition of the rules for selection of organizational units for the generalization of agility results in organizations to the Exemplar Agility Assessment Process and detailed evaluation of the exemplar process with more case studies.

REFERENCES

[1] R. T. Barry Boehm, "Balancing Agility and Discipline," in *Balancing Agility and Discipline – A Guide for the Perplexed*, ed: Addison Wesley, 2003.

[2] L. Williams and A. Cockburn, "Guest Editors' Introduction: Agile Software Development: It's about Feedback and Change," *Computer,* vol. 36, pp. 39-43, 2003.

[3] M. A. Mehdi Javanmard, "Comparison between Agile and Traditional software development methodologies," in *2nd National Conference on Applied Research inComputer Science and Information Technology*.

[4] M. Pikkarainen, O. Salo, R. Kuusela, and P. Abrahamsson, "Strengths and barriers behind the successful agile deployment--insights from the three software intensive companies in Finland," *Empirical Softw. Engg.,* vol. 17, pp. 675-702, 2012.

[5] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Commun. ACM,* vol. 48, pp. 72-78, 2005.

[6] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler*, et al.*, "The Agile Manifesto," 2001.

[7] K. Schwaber, "SCRUM Development Process," in *Business Object Design and Implementation: OOPSLA '95 Workshop Proceedings 16 October 1995, Austin, Texas*, J. Sutherland, C. Casanave, J. Miller, P. Patel, and G. Hollowell, Eds., ed London: Springer London, 1997, pp. 117-134.

[8] J. Stapleton, "DSDM: Dynamic Systems Development Method," presented at the Proceedings of the Technology of Object-Oriented Languages and Systems, 1999.

[9] M. Aoyama, "Agile Software Process model," presented at the Proceedings of the 21st International Computer Software and Applications Conference, 1997.

[10] A. Cockburn, *Writing effective use cases, The crystal collection for software professionals*: Addison-Wesley Professional Reading, 2000.

[11] K. Beck, *Extreme programming explained: embrace change*: Addison-Wesley Longman Publishing Co., Inc., 2000.

[12] I. James A. Highsmith, *Adaptive software development: a collaborative approach to managing complex systems*: Dorset House Publishing Co., Inc., 2000.

[13] S. R. Palmer and M. Felsing, *A Practical Guide to Feature-Driven Development*: Pearson Education, 2001.

[14] M. Poppendieck and T. Poppendieck, *Lean Software Development: An Agile Toolkit*: Addison-Wesley Longman Publishing Co., Inc., 2003.
[15] S. S. a. J. D. Arthur, "A Structured Framework for Assessing the “Goodness” of Agile Methods," presented at the 18th IEEE International Conference and Workshops on Engineering of Computer-Based Systems, 2011.
[16] B. B. Fraser S, Jarkvik J, Lundh E, Vilkki K. , "How do Agile/XP development methods affect companies?," presented at the 7th International Conference on Extreme Programming and Agile Processes in So ware Engineering, Oulu, Finland, 2006.
[17] H. Z. Taghi Javdani Gandomani, Abd Ghani, Abdul Azim, Md Sultan, Abu Bakar, "Important considerations for agile software development methods governance.," *Journal of Theoretical and Applied Information Technology,* vol. 55, pp. 345-351, 2013.
[18] J. D. A. Ahmed Sidky, "Value-Driven Agile Adoption: Improving An Organization's Software Development Approach," in *New Trends in Software Methodologies, Tools and Techniques* vol. 182, ed, 2008, pp. 149-164.
[19] C. P. a. M. Ramachandran, "Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices," *Int J Softw Eng,* pp. 3-28, 2009.
[20] A. J. Sidky A., "A Disciplined Approach to Adopting Agile Practices: The Agile Adoption Framework," *Innovations in Systems and Software Engineering 3,* pp. 203–216, 2007.
[21] S. W. Ambler. (2009, The Agile Scaling Model (ASM): Adapting Agile Methods for Complex En- vironments (2009). Available: ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/raw14204usen/RAW14204USEN.PDF
[22] R. Benefield, "Seven Dimensions of Agile Maturity in the Global Enterprise: A Case Study," in *2010 43rd Hawaii International Conference on System Sciences, HICSS*, 2010, pp. 1-7.
[23] A. Yin, Figueiredo, S., Mira da Silva, M., "Scrum Maturity Model: Validation for IT or- ganizations’ roadmap to develop software centered on the client role," in *The Sixth International Conference on Software Engineering Advances, ICSEA 2011*, 2011, pp. 10-29.
[24] H. Kniberg. *Scrum Checklist*. Available: https://www.crisp.se/gratis-material-och-guider/scrum-checklist
[25] J. Little. *Joe’s Unofficial Scrum Checklist*. Available: http://agileconsortium.pbworks.com/w/file/fetch/66642311/Joe%E2%80%99sUnofficialScrumCheckListV13.pdf
[26] E. Yatzeck. *A Corporate Agile 10-point Checklist*. Available: http://pagilista.blogspot.nl/2012/12/a-corporate-agile-10-point-checklist.html
[27] Storm-Consulting. *Storm-Consulting Agile Enterprise Survey*. Available: http://www.storm-consulting.com/agile-enterprise-survey/
[28] E. Ribeiro. *Agile Maturity Self-Assessment Survey*. Available: https://www.scrumalliance.org/community/articles/2015/december/agile-maturity-self-assessment-survey
[29] I. O. f. S. I. a. I. E. C. (IEC), "ISO/IEC 15504-3, 2004. ISO/IEC 15504-3:2004 - information technology - process assessment - part 3: guidance on performing an assessment," ed, 2004.

[30] P. Järvinen, *On Research Methods*, 2001.
[31] G. F. S. Salvatore T. March, "Design and Natural Science Research on Information Technology," *Decision Support Systems,* vol. 15, pp. 251-266, 1995.
[32] H. Z. Mina Ziaei Nafchi, Taghi Javdani Gandomani, "On The Current Agile Assessment Methods and Approaches," presented at the Malaysian Software Engineering Conference (MySEC), 2014.
[33] S. Hermida. *abetterteam.org*. Available: http://abetterteam.org/
[34] P. Thimmapuram. *Agile 3R Model of Maturity Assessment* Available: https://www.scrumalliance.org/community/articles/2015/march/agile-3r-model-maturity-assessment
[35] P. Nowinski. *Agile Assessment*. Available: https://nowinskipiotr.wordpress.com/2016/04/29/agile-assessment/
[36] Toughtworks. *Agile Assessments*. Available: http://www.agileassessments.com/
[37] I. Jacobson. *Agile Essentials*. Available: https://www.ivarjacobson.com/services/agile-essentials-starter-pack-agile-practices
[38] A. T. Inc. *Agile Health Assessment Tool*. Available: http://www.agilityhealthradar.com/teamhealth-assessment/ - teamhealth
[39] L. Lagestee. (2012). *Agile Health Dashboard*. Available: http://illustratedagile.com/2012/09/25/how-to-measure-team-agility/
[40] P. M. W. Krebs, R. Ashton. *Agile Journey Index*. Available: http://www.agiledimensions.com/
[41] Atlassian. *Agile Maturity Matrix in JIRA*. Available: http://blogs.atlassian.com/2013/11/agile-maturity-how-agile-is-your-organization/
[42] R. M. Iver. *Agile Maturity Self Assessment*. Available: http://www.robbiemaciver.com/documents/presentations/A2010-AgileMaturitySelf-Assessment.pdf
[43] I.-T. R. Group. *Agile Process Assessment Tool*. Available: http://www.infotech.com/research/ss/it-deploy-changes-more-rapidly-by-going-agile/it-agile-process-assessment-tool
[44] P. Gadget. *Agile Readiness Questionnaire*. Available: http://www.pmgadget.co.uk/agile-readiness-questionnaire
[45] C. P. Management. *Agile Self Assessment*. Available: http://www.agileprojectmanagementtraining.com/agile-self-assessment/
[46] M. Hoogveld. *Agile Self Assessment*. Available: http://www.mikehoogveld.nl/ - !agility-assessment/nxo8j
[47] E. Gunnerson. *Agile team evaluation*. Available: http://blogs.msdn.com/b/ericgu/archive/2015/10/05/agile-team-evaluation.aspx
[48] M. Britsch. (2014). *Agility Questionnaire*. Available: http://www.thedigitalbusinessanalyst.co.uk/2014/07/Agile-Questionnaire.html
[49] S. Matalonga, "AGIS " *Computación y Sistemas,* vol. 15, 2015.
[50] C. Schwaber. (2007) The Truth About Agile Proceses - Frank Answers to Frequently Asked Questions. *Forrester Research*.

[51] A. K. V. Ameya S. Erande, "Measuring Agility of Organizations – A Comprehensive Agility Measurement Tool (CAMT)," in *IAJC-IJME International Conference*, 2008.
[52] M. Sahota. (2012). *An Agile Adoption and Transformation Survival Guide*.
[53] e. a. L. Williams. *Comparative Agility*. Available: http://comparativeagility.com/
[54] C. Achouiantz. (2013). *Depth of Kanban*. Available: http://leanagileprojects.blogspot.se/2013/03/depth-of-kanban-good-coaching-tool.html
[55] DrAgile. *Enterprise Agile Practice Assessment Tool* Available: http://www.dragile.com/
[56] E. Group. (2013). *Introducing the Agility Maturity Matrix*. Available: http://www.eliassen.com/agile-blog/introducing-the-enterprise-agility-maturity-matrix
[57] M. V. Diego Lo Giudice, Phil and R. C. Murphy. *Forrester's Agile Testing Maturity Assessment Tool*. Available: https://www.forrester.com/report/Forresters+Agile+Testing+Maturity+Assessment+Tool/-/E-RES90341
[58] O. R. Yürüm, "GSPA: A GENERIC SOFTWARE PROCESS ASSESSMENT TOOL," Master of Science, The Graduate School of Informatics, Middle East Technical University, 2014.
[59] K. Waters. *How Agile Are You?* . Available: http://www.allaboutagile.com/how-agile-are-you-take-this-42-point-test/
[60] IBM. *IBM DevOps*. Available: http://www.ibm.com/ibm/devops/us/en/
[61] L. A. I. (LAI). *LAI Enterprise Self-Assessment Tool* Available: http://hdl.handle.net/1721.1/84688
[62] M. P. Kumar. *Maturity Assessment Model for Scrum Teams*. Available: https://www.scrumalliance.org/community/articles/2014/july/maturity-assessment-model-for-the-scrum-teams
[63] NetObjectives. *Net Objectives Lean-Agile Roadmap for Achieving Enterprise Agility*. Available: http://www.netobjectives.com/resources/lean-agile-roadmap
[64] S. Inc. *Open Assessments* Available: https://www.scrum.org/Assessments/Open-Assessments
[65] S. Miller. *Readiness & Fit Analysis* Available: https://insights.sei.cmu.edu/sei_blog/2012/10/readiness-fit-analysis.html
[66] S. Elleithy. *ReadyForAgile*. Available: http://agilepreview.com/
[67] D. Hawks. *Scrum Assessment Series*. Available: http://www.agilevelocity.com/scrum-assessment-series/
[68] J. S. Bas Vodde. *Scrum Butt Test (Nokia Test)*. Available: https://www.scruminc.com/scrum-blog/
[69] B. Gloger. (2012). *Scrum Checklist 2012*. Available: http://www.infoq.com/minibooks/scrum-checklists
[70] C. P. Disorn Homchuenchom, Horst Lichter, Toni Anwar, "SPIALS: A light-weight Software Process Improvement Self-Assessment Tool," presented at the 5th Malaysian Conference in Software Engineering (MySEC), 2011.
[71] H. Kniberg. *Squad Health Check Model*. Available: https://labs.spotify.com/2014/09/16/squad-health-check-model/

[72] J. Janlén. (2014). *Team Barometer (Self-Evaluation Tool)* Available: http://blog.crisp.se/2014/01/30/jimmyjanlen/team-barometer-self-evaluation-tool
[73] C. Verwijs. *TeamMetrics*. Available: http://teammetrics.apphb.com/
[74] H. Kniberg. *The Unofficial Scrum Checklist*. Available: https://www.crisp.se/file-uploads/scrum-checklist.pdf
[75] D. Leffingwell, *Scaling Software Agility: Best Practices for Large Enterprises (The Agile Software Development Series)*: Addison-Wesley Professional, 2007.
[76] W. Krebs, "Level up Your Agile with the Agile Journey Index," ed, 2013.
[77] M. Doyle, Williams, L., Cohn, M., & Rubin, K. S., "Agile software development in practice," *In Agile Processes in Software Engineering and Extreme Programming,* pp. 32-45, 2014.
[78] . *Comparative Agility Overview*. Available: https://comparativeagility.com/overview
[79] D. Anderson, *Kanban - Successful Evolutionary Change for your Technology Business*: Blue Hole Press, 2010.
[80] E. Group. (2013, Enterprise Agility Guide. Available: https://www.dropbox.com/sh/hxeoumi1rhzj1or/WIvY-fX0U7/EnterpriseAgility
[81] Atlassian. *Jira Software*. Available: https://www.atlassian.com/software/jira
[82] R. Kneuper, *CMMI: Improving Software and Systems Development Processes Using Capability Maturity Model Integration*: Rocky Nook, 2008.
[83] I. O. f. S. I. a. I. E. C. (IEC), "ISO/IEC 15504-2:2003 Information technology -- Process assessment -- Part 2: Performing an assessment," ed, 2003.
[84] O. Ozcan-Top, "A Reference Model for Software Agility Assessment: AgilityMod " PhD, Middle East Technical University, 2014.
[85] O. Ozcan-Top and O. Demirörs, "A Reference Model for Software Agility Assessment: AgilityMod," in *Software Process Improvement and Capability Determination*. vol. 526, T. Rout, R. V. O'Connor, and A. Dorling, Eds., ed: Springer International Publishing, 2015, pp. 145-158.
[86] Scrum.org. *Scrum.org*. Available: https://www.scrum.org/
[87] J. R. Hackman and G. R. Oldham, "Development of the Job Diagnostic Survey," *Journal of Applied Psychology,* vol. 60, pp. 159-170, 1975.
[88] A. B. B. Wilmar B. Schaufeli, Marisa Salanova, "The Measurement of Work Engagement With a Short Questionnaire," *Educational and Psychological Measurement,* vol. 66, pp. 701-716, 2006.
[89] *Agile Manifesto*, 2001.
[90] S. W. Ambler and M. Lines, *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*: IBM Press, 2012.
[91] IEEE, "IEEE Recommended Practice for Software Requirements Specifications," *IEEE Std 830-1998,* pp. 1-40, 1998.
[92] "IEEE Standard for Information Technology--Systems Design--Software Design Descriptions - Redline," *IEEE Std 1016-2009 (Revision of IEEE Std 1016-1998) - Redline,* pp. 1-58, 2009.
[93] . *Bootstrap Front-end Framework*. Available: getbootstrap.com

[94] I. O. f. S. I. a. I. E. C. (IEC), "ISO/IEC 15504-5:2012 Information technology -- Process assessment -- Part 5: An exemplar software life cycle process assessment model," ed, 2012.
[95] A. M. Lund, "Measuring Usability with the USE Questionnaire," *STC Usability SIG Newsletter,* 2001.
[96] R. K. Yin, *Case Study Research: Design and Methods*, Fifth Edition ed.: SAGE Publications, 2014.

APPENDIX A

USE CASES

Add Organization

Table 29: Add Organization Use Case Scenario

Use Case:	1: Add Organization
Scope	AssessAgility
Priority	Essential
Description	Admin wants to add an organization.
Primary Actor	Admin
Precondition(s)	Admin is identified and authenticated.
Main Success Scenario (MSS)	1. Admin navigates to the "Organizations" pane. 2. System displays the list of the organizations. 3. Admin clicks to the "New Organization" button from the "Floating Action Button". 4. System loads the "Create Organization" page. 5. Admin enters the information about the organization that he/she wants to add and clicks "Create" button. 6. System saves the new organization, returns to the "Organizations" pane and displays the list of organizations.

Extensions	None
Post condition(s)	An organization record is generated and recorded into the database.
Exceptions	5.a. The information given by Admin is incorrect or missing. 1. System displays error messages beneath the related fields. 2. MSS continues with step 5.
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 15: AssessAgility Organizations Page

Edit Organization

Table 30: Edit Organization Use Case Scenario

Use Case:	2: Edit Organization
Scope	AssessAgility
Priority	Essential
Description	Admin wants to update the information of an existing organization.
Primary Actor	Admin
Precondition(s)	Admin is identified and authenticated.
Main Success Scenario (MSS)	1. Admin navigates to the "Organizations" pane. 2. System displays the list of the organizations. 3. Admin selects the organization that he/she wants to edit by clicking the "Edit" button. 4. System loads the "Edit Organization" page and displays information about the selected organization. 5. Admin changes the information of the organization and clicks "Save" button.

	6. System updates the organization information, returns to the "Organizations" pane and displays the list of organizations.
Extensions	None
Post condition(s)	Related organization record is updated and recorded into the database.
Exceptions	5.a. The information given by Admin is incorrect or missing. 1. System displays error messages beneath the related fields. 2. MSS continues with step 5.
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 15: AssessAgility Organizations Page

Delete Organization

Table 31: Delete Organization Use Case Scenario

Use Case:	3: Delete Organization
Scope	AssessAgility
Priority	Essential
Description	Admin wants to delete an existing organization.
Primary Actor	Admin
Precondition(s)	Admin is identified and authenticated.
Main Success Scenario (MSS)	1. Admin navigates to the "Organizations" pane. 2. System displays the list of the organizations. 3. Admin selects the organization that he/she wants to delete by clicking the "Delete" button. 4. System loads the "Delete

	Organization" page, displays information about the selected organization and asks "Are you sure you want to delete this organization?" 5. Admin clicks the "Delete" button. 6. System deletes the organization and users related to the organization, returns to the "Organizations" pane and displays the list of organizations.
Extensions	5.a. Admin clicks to the "Back to List" button. 1. MSS continues with step 2.
Post condition(s)	The selected organization record and all the users related to that organization is deleted from the database.
Exceptions	None
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 15: AssessAgility Organizations Page

Add User

Table 32: Add User Use Case Scenario

Use Case:	4: Add User
Scope	AssessAgility
Priority	Essential
Description	Admin wants to add a user.
Primary Actor	Admin
Precondition(s)	Admin is identified and authenticated.
Main Success Scenario (MSS)	1. Admin navigates to the "Users" pane. 2. System displays the list of the users. 3. Admin clicks to the "New User" button from the "Floating Action Button". 4. System loads the "Create User" page.

	5. Admin enters the information about the user that he/she wants to add and clicks "Create" button. 6. System saves the new user, returns to the "Users" pane and displays the list of users.
Extensions	None
Post condition(s)	A user record is generated and recorded into the database.
Exceptions	5.a. The information given by Admin is incorrect or missing. 1. System displays error messages beneath the related fields. 2. MSS continues with step 5.
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 16: AssessAgility Users Page

Edit User

Table 33: Edit User Use Case Scenario

Use Case:	5: Edit User
Scope	AssessAgility
Priority	Essential
Description	Admin wants to update the information of an existing user.
Primary Actor	Admin
Precondition(s)	Admin is identified and authenticated.
Main Success Scenario (MSS)	1. Admin navigates to the "Users" pane. 2. System displays the list of the users. 3. Admin selects the user that he/she wants to edit by clicking the "Edit" button.

	4. System loads the “Edit User” page and displays information about the selected user. 5. Admin changes the information of the user and clicks “Save” button. 6. System updates the user information, returns to the “Users” pane and displays the list of users.
Extensions	None
Post condition(s)	Related user record is updated and recorded into the database.
Exceptions	5.a. The information given by Admin is incorrect or missing. 1. System displays error messages beneath the related fields. 2. MSS continues with step 5.
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 16: AssessAgility Users Page

Delete User

Table 34: Delete User Use Case Scenario

Use Case:	6: Delete User
Scope	AssessAgility
Priority	Essential
Description	Admin wants to delete an existing organization.
Primary Actor	Admin
Precondition(s)	Admin is identified and authenticated.
Main Success Scenario (MSS)	1. Admin navigates to the “Users” pane. 2. System displays the list of the organizations. 3. Admin selects the user that he/she

	wants to delete by clicking the "Delete" button. 4. System loads the "Delete User" page, displays information about the selected organization and asks "Are you sure you want to delete this user?" 5. Admin clicks the "Delete" button. 6. System deletes the organization, returns to the "Users" pane and displays the list of users.
Extensions	5.a. Admin clicks to the "Back to List" button. 1. MSS continues with step 2.
Post condition(s)	Selected user record is deleted from the database.
Exceptions	None
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 16: AssessAgility Users Page

Create Project

Table 35: Create Project Use Case Scenario

Use Case:	7: Create Project
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to create a project.
Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated.
Main Success Scenario (MSS)	1. Lead Assessor navigates to the "Projects" pane. 2. System displays the list of the projects. 3. Lead Assessor clicks to the "New Project" button from the "Floating

	Action Button". 4. System loads the "Create Project" page. 5. Lead Assessor enters the information about the project and clicks to the "Create" button. 6. System saves the new project, returns to the "Projects" pane and displays the list of projects.
Extensions	None
Post condition(s)	A project record is generated and recorded into the database.
Exceptions	5.a. The information given by Lead Assessor is incorrect or missing. 1. System displays error messages beneath the related fields. 2. MSS continues with step 5.
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 17: AssessAgility Projects Page

Edit Project

Table 36: Edit Project Use Case Scenario

Use Case:	8: Edit Project
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to edit a project.
Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated.
Main Success Scenario (MSS)	1. Lead Assessor navigates to the "Projects" pane. 2. System displays the list of the projects.

	3. Lead Assessor selects the project that he/she wants to edit by clicking the "Edit" button. 4. System loads the "Edit Project" page and displays information about the selected project. 5. Lead Assessor changes the information of the project and clicks "Save" button. 6. System updates the project information, returns to the "Projects" pane and displays the list of projects.
Extensions	None
Post condition(s)	A project record is updated and recorded into the database.
Exceptions	5.a. The information given by Lead Assessor is incorrect or missing. 1. System displays error messages beneath the related fields. 2. MSS continues with step 5.
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 17: AssessAgility Projects Page

Delete Project

Table 37: Delete Project Use Case Scenario

Use Case:	9: Delete Project
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to delete an existing project.
Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated.
Main Success Scenario	1. Lead Assessor navigates to the

(MSS)	"Projects" pane. 2. System displays the list of the projects. 3. Lead Assessor selects the project that he/she wants to delete by clicking the "Delete" button. 4. System loads the "Delete Project" page, displays information about the selected team and asks "Are you sure you want to delete this project?" 5. Lead Assessor clicks the "Delete" button. 6. System deletes the project, removes the assessments for that project and displays the list of projects.
Extensions	5.a. Admin clicks to the "Back to List" button. 1. MSS continues with step 2.
Post condition(s)	Selected project record, all the assessments made for that project and assignments between the deleted project and assigned teams are deleted from the database.
Exceptions	None
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 17: AssessAgility Projects Page

Create Team

Table 38: Create Team

Use Case:	10: Create Team
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to create a team.

Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated. There must be at least one assessor defined in the system.
Main Success Scenario (MSS)	1. Lead Assessor navigates to the "Teams" pane. 2. System displays the list of the teams. 3. Lead Assessor clicks to the "New Team" button from the "Floating Action Button". 4. System loads the "Create Team" page. 5. Lead Assessor enters the information about the team, add team members and clicks to the "Create" button. 6. System saves the new team, returns to the "Teams" pane, displays the list of teams and sends notification e-mails to the team members about their involvement in the team.
Extensions	None
Post condition(s)	A team record is generated and recorded into the database.
Exceptions	5.a. The information given by Lead Assessor is incorrect or missing. 1. System displays error messages beneath the related fields. 2. MSS continues with step 5.
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 18: AssessAgility Teams Page

Edit Team

Table 39: Create Team Use Case Scenario

Use Case:	11: Edit Team
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to edit a team.
Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated.
Main Success Scenario (MSS)	1. Lead Assessor navigates to the "Teams" pane. 2. System displays the list of the teams. 3. Lead Assessor selects the team that he/she wants to edit by clicking the "Edit" button. 4. System loads the "Edit Team" page and displays information about the selected team. 5. Lead Assessor changes the information of the team and clicks "Save" button. 6. System updates the team information, returns to the "Teams" pane, displays the list of teams and sends notification e-mails to the team members about the update.
Extensions	None
Post condition(s)	A team record is updated and recorded into the database.
Exceptions	5.a. The information given by Lead Assessor is incorrect or missing. 1. System displays error messages beneath the related fields. 2. MSS continues with step 5.
Special Requirements	None

Reference	Figure 4: UCD Diagram Figure 18: AssessAgility Teams Page

Delete Team

Table 40: Delete Team Use Case Scenario

Use Case:	12: Delete Team
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to delete an existing team.
Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated.
Main Success Scenario (MSS)	1. Lead Assessor navigates to the "Teams" pane. 2. System displays the list of the teams. 3. Lead Assessor selects the team that he/she wants to delete by clicking the "Delete" button. 4. System loads the "Delete Team" page, displays information about the selected team and asks "Are you sure you want to delete this team?" 5. Lead Assessor clicks the "Delete" button. 6. System deletes the team, removes the affiliations of the team members returns to the "Teams" pane, displays the list of teams and sends notification e-mails to the team members.
Extensions	5.a. Lead Assessor clicks to the "Back to List" button. 1. MSS continues with step 2.
Post condition(s)	Selected team record and all the relationship between team and its members are deleted from the database.

Exceptions	None
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 18: AssessAgility Teams Page

Make Assignment

Table 41: Make Assignment Use Case Scenario

Use Case:	13: Make Assignment
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to make an assignment.
Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated. There must be at least one assessor defined in the system.
Main Success Scenario (MSS)	1. Lead Assessor navigates to the "Assignments" pane. 2. System displays the list of the assignments. 3. Lead Assessor clicks to the "New Assignment" button from the "Floating Action Button". 4. System loads the "Make Assignment" page. 5. Lead Assessor chooses the project, team, defines assignments between team members and assigns a due date for the assignment. 6. System saves the new assignment, returns to the "Assignments" pane, displays the list of assignments and sends notification e-mails to the team

	members about their assignment.
Extensions	None
Post condition(s)	An assignment record is generated and recorded into the database.
Exceptions	5.a. The information given by Lead Assessor is incorrect or missing. 1. System displays error messages beneath the related fields. 2. MSS continues with step 5.
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 20: AssessAgility Assignments Page Figure 21: AssessAgility Make Assignment Page

Edit Assignment

Table 42: Edit Assignment Use Case Scenario

Use Case:	14: Edit Assignment
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to edit an assignment.
Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated.
Main Success Scenario (MSS)	1. Lead Assessor navigates to the "Assignments" pane. 2. System displays the list of the assignments. 3. Lead Assessor selects the assignment that he/she wants to edit by clicking the "Edit" button. 4. System loads the "Edit Assignment" page and displays information about the selected assignment.

	5. Lead Assessor changes the information of the assignment and clicks "Save" button. 6. System updates the assignment information, returns to the "Assignments" pane, displays the list of assignments and sends notification e-mails to the team members about the update.
Extensions	None
Post condition(s)	An assignment record is updated and recorded into the database.
Exceptions	5.a. The information given by Lead Assessor is incorrect or missing. 1. System displays error messages beneath the related fields. 2. MSS continues with step 5.
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 20: AssessAgility Assignments Page

Undo Assignment

Table 43: Undo Assignment Use Case Scenario

Use Case:	15: Undo Assignment
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to delete an existing assignment.
Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated.
Main Success Scenario (MSS)	1. Lead Assessor navigates to the "Assignments" pane. 2. System displays the list of the

	assignments. 3. Lead Assessor selects the assignment that he/she wants to delete by clicking the "Delete" button. 4. System loads the "Delete Assignment" page, displays information about the selected team and asks "Are you sure you want to delete this assignment?" 5. Lead Assessor clicks the "Delete" button. 6. System deletes the assignment, removes the affiliations of the team members returns to the "Assignments" pane, displays the list of assignments and sends notification e-mails to the team members.
Extensions	5.a. Lead Assessor clicks to the "Back to List" button. 1. MSS continues with step 2.
Post condition(s)	Selected assignment record and all the relationship between previously assigned team members are deleted from the database.
Exceptions	None
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 20: AssessAgility Assignments Page

Analyze Assessment

Table 44: Analyze Assessment Use Case Scenario

Use Case:	16: Analyze Assessment
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to analyze an assessment.

Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated. There has to be at least one assessment.
Main Success Scenario (MSS)	1. Lead Assessor navigates to the "Assessments" pane. 2. System displays the list of the assessments. 3. Lead Assessor selects a completed assessment from the list. 4. System loads the "Analyze Assessment" page. 5. Lead Assessor analyzes the assessment and concludes the assessment. 6. System saves the changes on the assessment, returns to the "Assessments" pane and displays the list of assessments.
Extensions	5.a. Lead Assessor approves the assessment. 1. System changes the status of the assessment from "Completed" to "Approved" and send a notification e-mail to the parties assigned to the assessment. 2. MSS continues with step 6. 5.b. Lead Assessor rejects the assessment. 1. System changes the status of the assessment from "Completed" to "Rejected" and send a notification e-mail to the parties assigned to the assessment. 2. MSS continues with step 6.
Post condition(s)	Selected assessment is updated.
Exceptions	None
Special Requirements	None
Reference	Figure 4: UCD Diagram

	Figure 22: AssessAgility Analyze Assessment Page

Generate Single Report

Table 45: Generate Single Report Use Case Scenario

Use Case:	17: Generate Single Report
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to generate a report for an assessment.
Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated. There has to be at least one completed assessment.
Main Success Scenario (MSS)	1. Lead Assessor navigates to the "Reports" pane. 2. System displays the list of the assessments. 3. Lead Assessor selects a completed assessment from the list by clicking on "Get Report" button. 4. System generates the assessment report.
Extensions	None
Post condition(s)	A report is generated for the selected assessment.
Exceptions	None
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 23: AssessAgility Reports Page

Generate Combination Report

Table 46: Generate Combination Report Use Case Scenario

Use Case:	18: Generate Combination Report
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to generate a report for multiple assessments combined.
Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated. There has to be at least two completed assessment.
Main Success Scenario (MSS)	1. Lead Assessor navigates to the "Reports" pane. 2. System displays the list of the reports. 3. Lead Assessor clicks on the "Combine Assessments" button from the "Floating Action Button" 4. System loads "Combine Assessments" page. 5. Lead Assessor chooses the assessments that he/she wants to get a combined report for and clicks "Get Report" button. 6. System generates the combined report.
Extensions	None
Post condition(s)	A combined report is generated for the selected assessments.
Exceptions	None
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 23: AssessAgility Reports Page

Generate Comparison Report

Table 47: Generate Comparison Report Use Case Scenario

Use Case:	19: Generate Comparison Report
Scope	AssessAgility
Priority	Essential
Description	Lead Assessor wants to generate a comparison report for multiple assessments.
Primary Actor	Lead Assessor
Precondition(s)	Lead Assessor is identified and authenticated. There has to be at least two completed assessment.
Main Success Scenario (MSS)	1. Lead Assessor navigates to the "Reports" pane. 2. System displays the list of the reports. 3. Lead Assessor clicks on the "Compare Assessments" button from the "Floating Action Button" 4. System loads "Compare Assessments" page. 5. Lead Assessor chooses the assessments that he/she wants to get a comparison report for and clicks "Get Report" button. 6. System generates the comparison report.
Extensions	None
Post condition(s)	A comparison report is generated for the selected assessments.
Exceptions	None
Special Requirements	Up to 5 reports can be chosen for comparison.

Reference	Figure 4: UCD Diagram Figure 23: AssessAgility Reports Page Figure 24: AssessAgility Compare Reports Page

Perform Assessment

Table 48: Perform Assessment Use Case Scenario

Use Case:	20: Perform Assessment
Scope	AssessAgility
Priority	Essential
Description	Assessor wants to perform an assessment that is assigned him/her by Lead Assessor.
Primary Actor	Assessor
Precondition(s)	Assessor is identified and authenticated. There has to be at least one assignment made for the Assessor.
Main Success Scenario (MSS)	1. Assessor navigates to the "Assessments" pane. 2. System displays the list of the assessments. 3. Assessor selects the assessment that he/she wants to perform by clicking on the "Perform" button. 4. System loads "Perform Assessment" page. 5. Assessor fills in the assessment. 6. Assessor ends the performing session. 7. System records the assessment, returns to the "Assessments" pane.
Extensions	6.a. Assessor pauses the assessment. 1. System changes assessment's status to "In Progress".

	2. MSS continues with step 7. 6.b. Assessor finishes the assessment. 1. System changes assessment's status to "Completed" and send notification to the related Lead Assessor. 2. MSS continues with step 7.
Post condition(s)	All progress related to the assessment is recorded.
Exceptions	None
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 25: AssessAgility Perform Assessment Page

Register

Table 49: Register Use Case Scenario

Use Case:	21: Register
Scope	AssessAgility
Priority	Essential
Description	Self-Assessor wants to register to the System.
Primary Actor	Self-Assessor
Precondition(s)	None
Main Success Scenario (MSS)	1. Self-Assessor clicks to the "Register" button on the home screen. 2. System displays the "Registry Form". 3. Self-Assessor enters the required information. 4. System checks the information entered. 5. System generates an account for the Self-Assessor and returns to the home screen.

Extensions	None
Post condition(s)	An account is created for the Self-Assessor.
Exceptions	4.a. Given information is missing or invalid. 1. System highlights the related fields and displays error messages at the "Registry Form" 2. MSS continues with step 3.
Special Requirements	All self-assessors are required to have valid and unique e-mail addresses. Passwords must be at least 6 characters long and must contain an uppercase letter, a lower case letter, a number and a symbol.
Reference	Figure 4: UCD Diagram Figure 14: AssessAgility Home Page

Login

Table 50: Login Use Case Scenario

Use Case:	22: Login
Scope	AssessAgility
Priority	Essential
Description	User wants to log in to the System.
Primary Actor	User
Precondition(s)	User has an account registered to the System.
Main Success Scenario (MSS)	1. User clicks to the "Login" button on the home screen. 2. System displays the "Login Form". 3. User enters his/hers e-mail and password. 4. System checks the information entered.

	5. System loads the convenient main screen according to the type of the User.
Extensions	None
Post condition(s)	User is logged in to the System.
Exceptions	4.a. Given information is missing or invalid. 3. System highlights the related fields and displays error messages at the "Login Form" 4. MSS continues with step 3.
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 14: AssessAgility Home Page

Update Profile

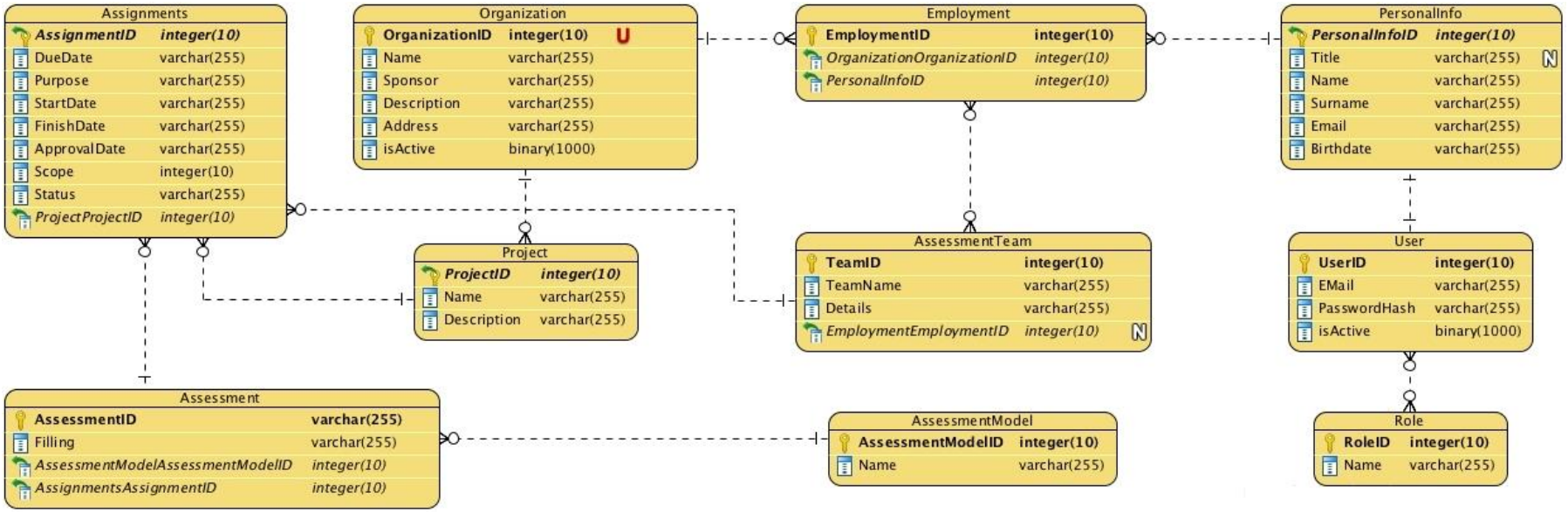
Table 51: Update Profile Use Case Scenario

Use Case:	23: Update Profile
Scope	AssessAgility
Priority	Essential
Description	User wants to update the information in his profile.
Primary Actor	User
Precondition(s)	User is identified and authenticated.
Main Success Scenario (MSS)	1. User clicks to the "Profile" button on the home screen. 2. System loads the "View Profile" page and displays the profile information to the user.

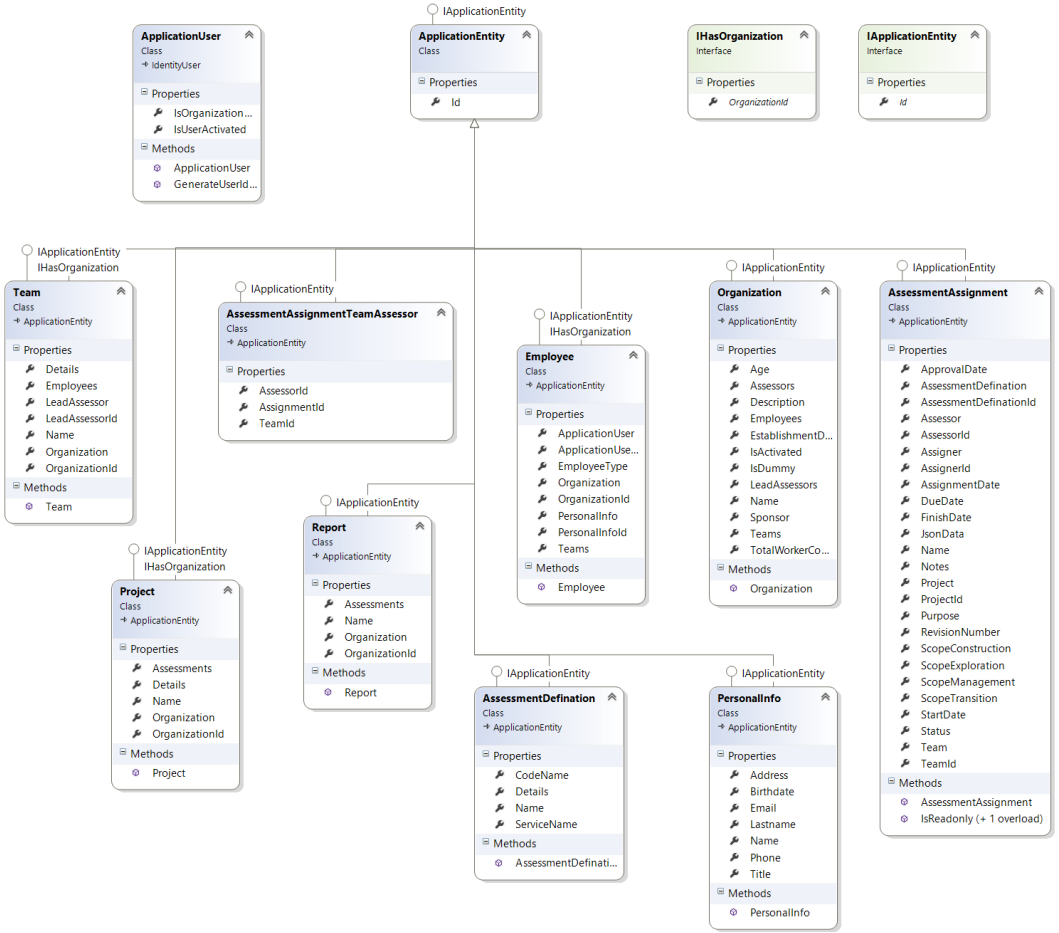
	3. User changes the information on his/her profile. 4. System checks the information entered. 5. System returns to the "View Profile" page and displays the profile information to the user.
Extensions	None
Post condition(s)	User profile is updated.
Exceptions	4.a. Given information is missing or invalid. 5. System highlights the related fields and displays error messages at the "View Profile" page. 6. MSS continues with step 3.
Special Requirements	None
Reference	Figure 4: UCD Diagram Figure 26: AssessAgility Update Profile Page

APPENDIX B

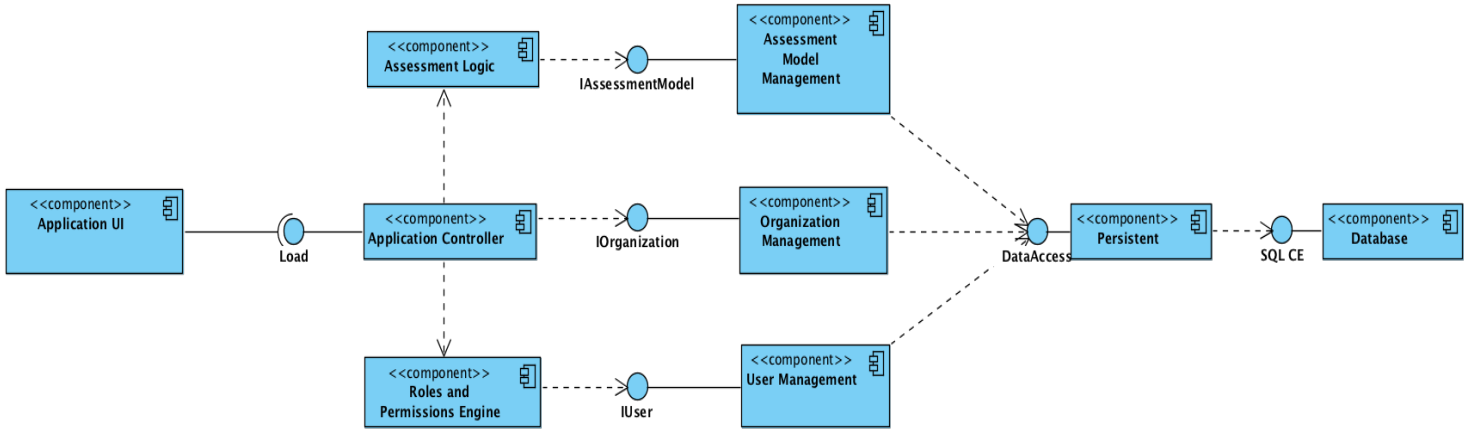
ER DIAGRAM



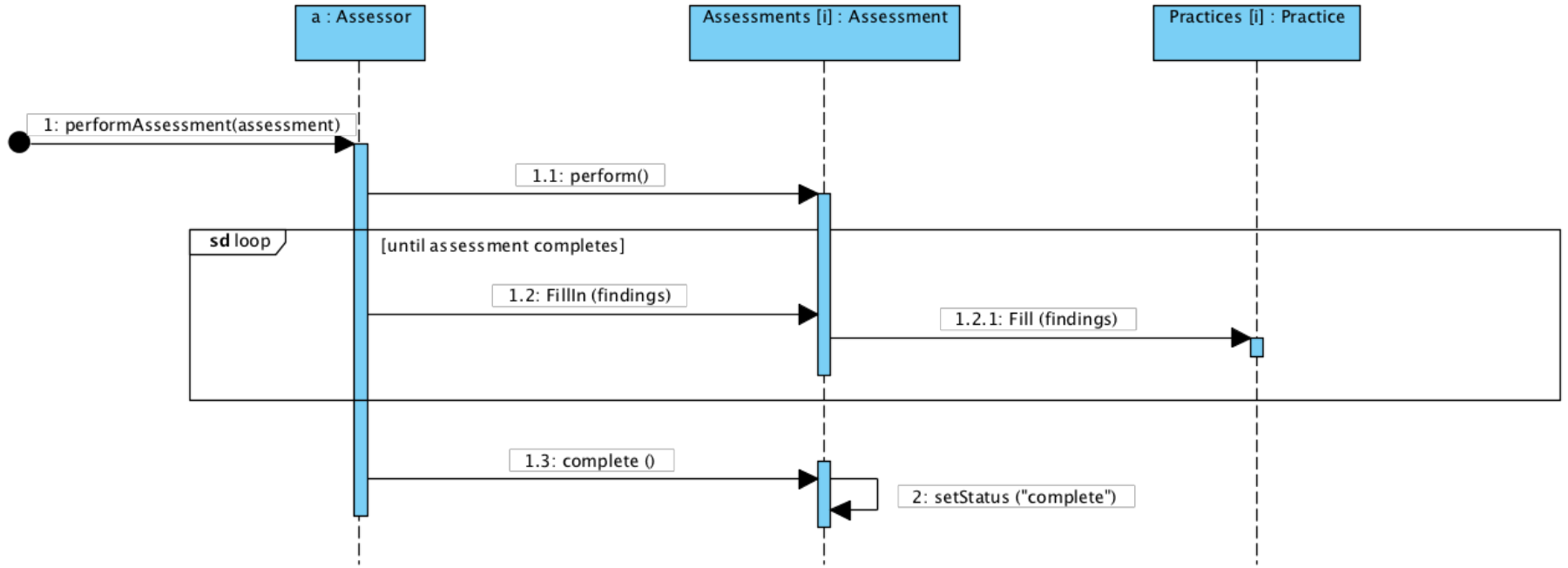
CLASS DIAGRAM



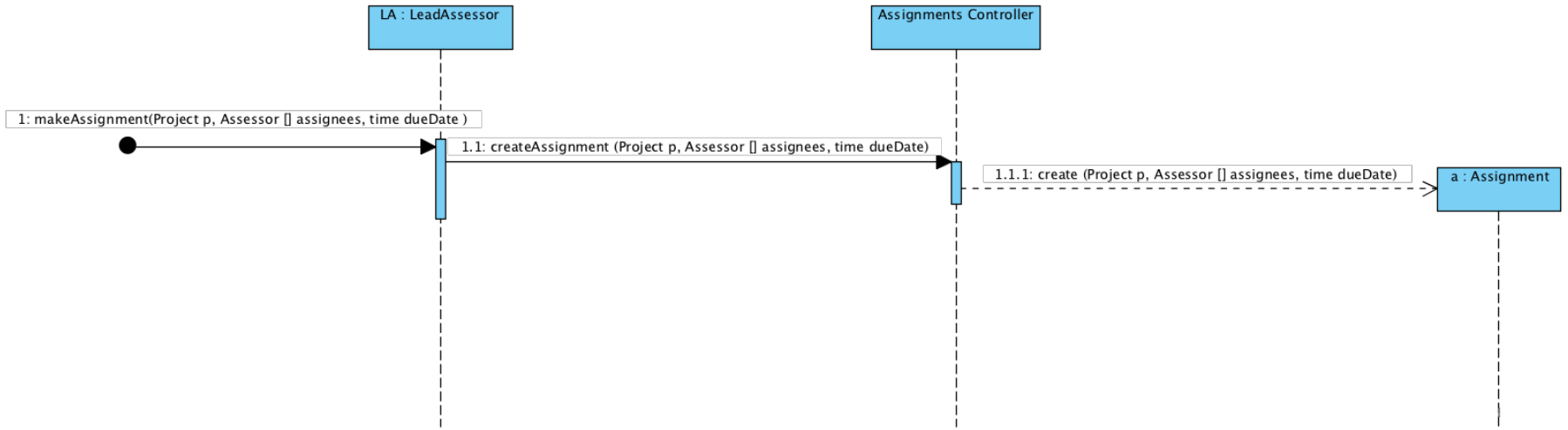
COMPONENT DIAGRAM



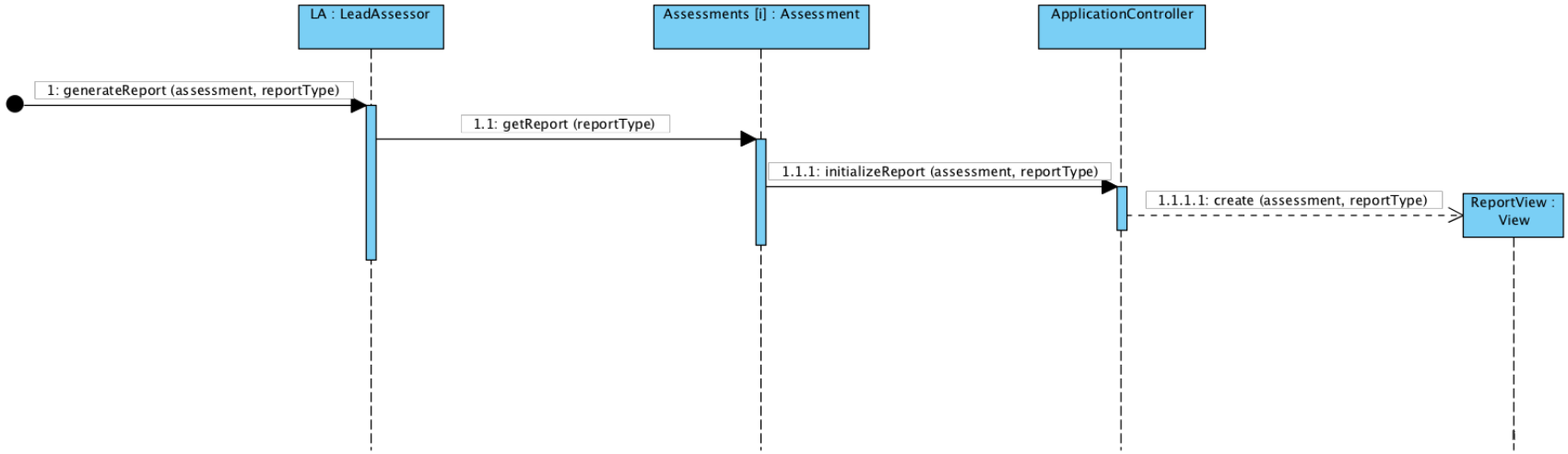
PERFORM ASSESSMENT SEQUENCE DIAGRAM



MAKE ASSESSMENT SEQUENCE DIAGRAM



GENERATE REPORT SEQUENCE DIAGRAM



APPENDIX C

SCREENSHOTS OF THE TOOL

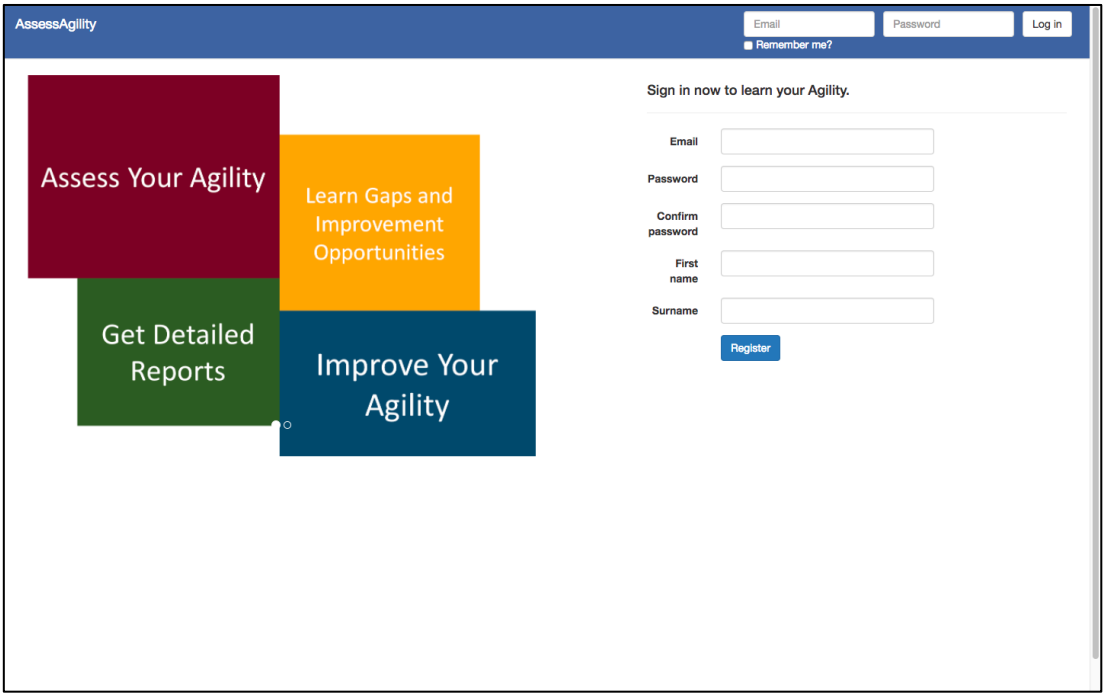


Figure 14: AssessAgility Home Page

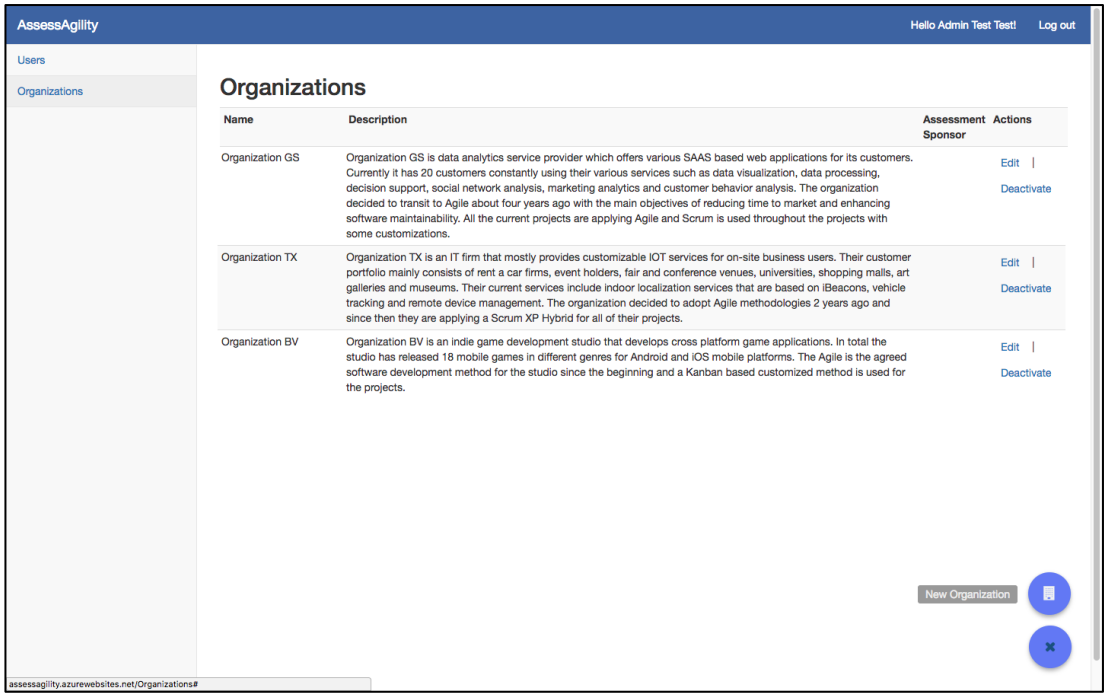


Figure 15: AssessAgility Organizations Page

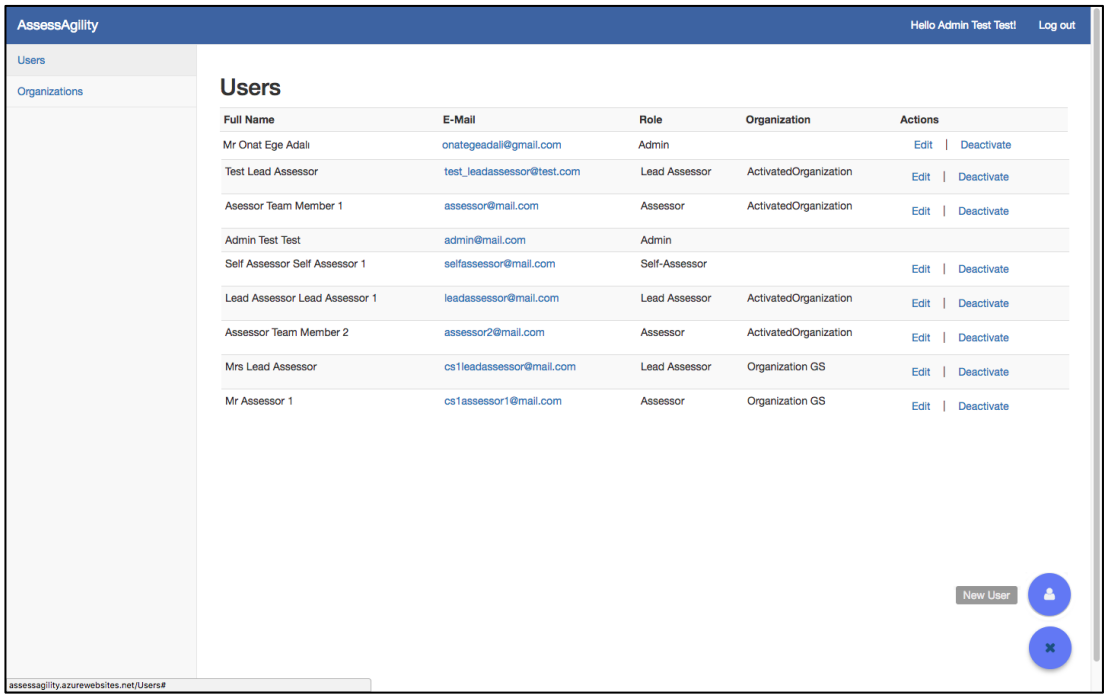


Figure 16: AssessAgility Users Page

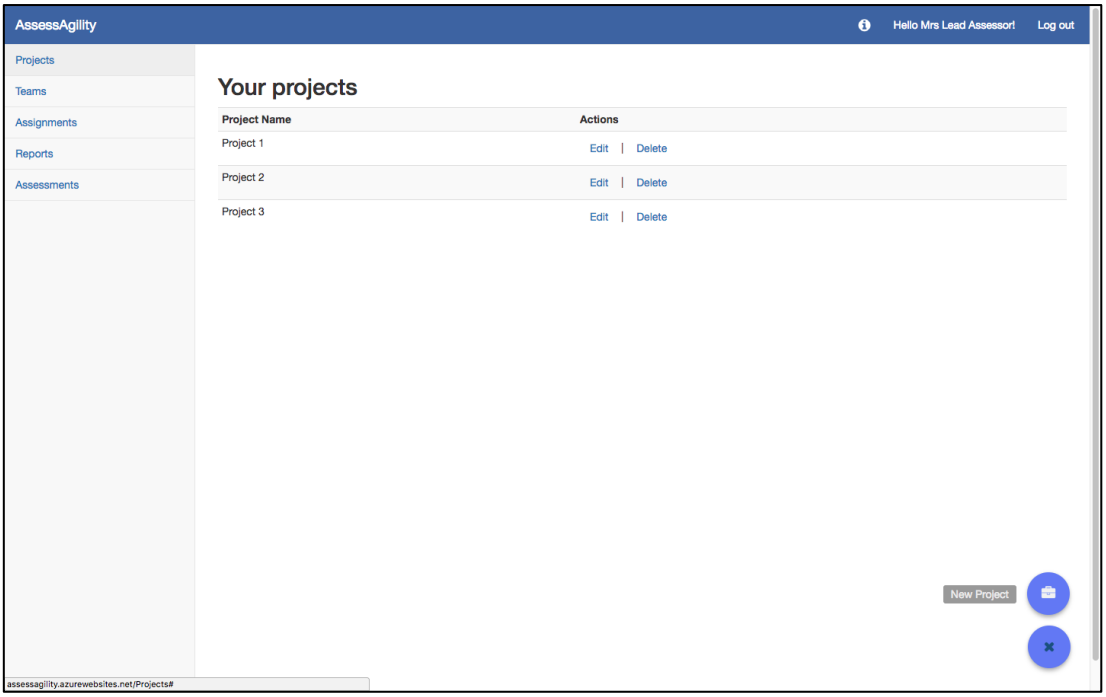


Figure 17: AssessAgility Projects Page

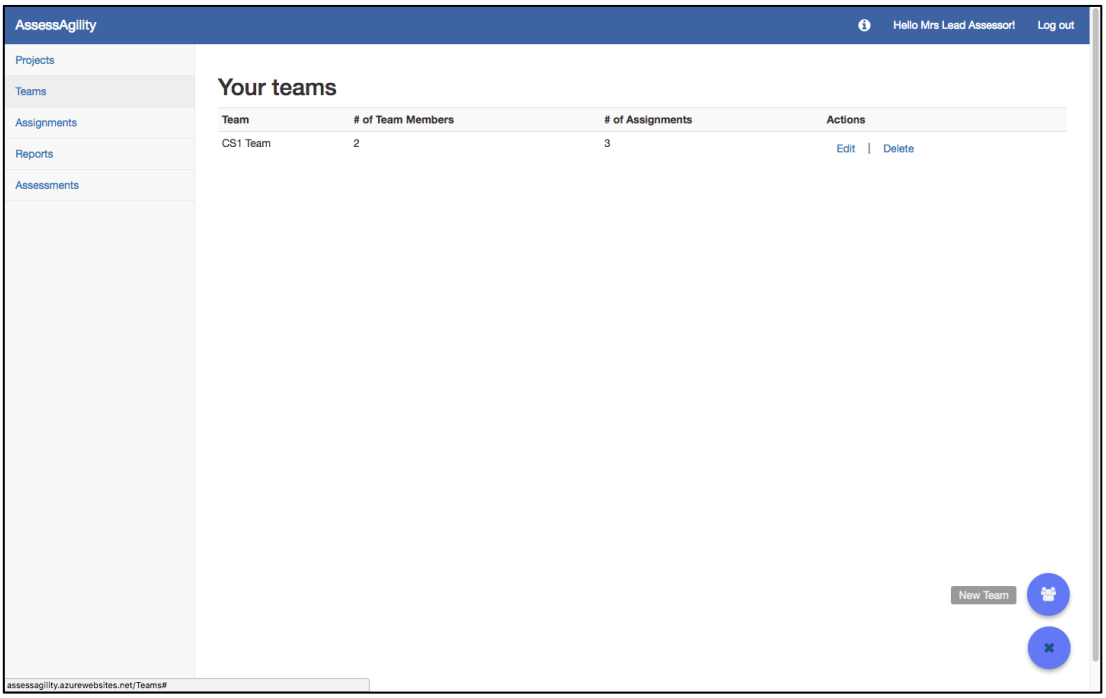


Figure 18: AssessAgility Teams Page

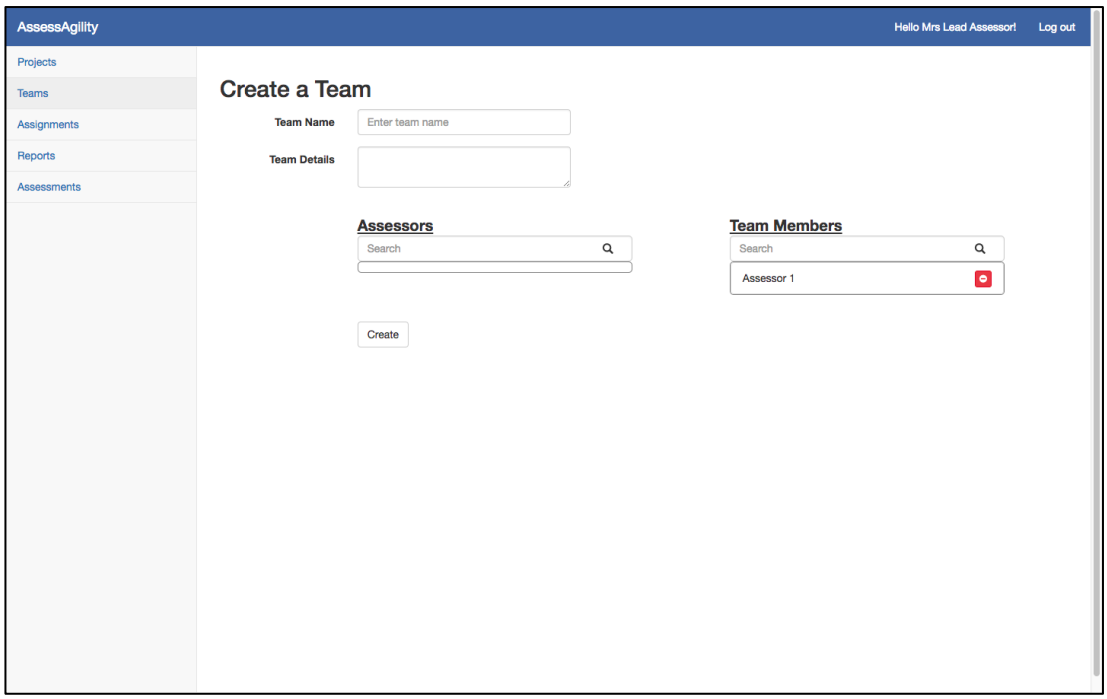


Figure 19: AssessAgility Create Team Page

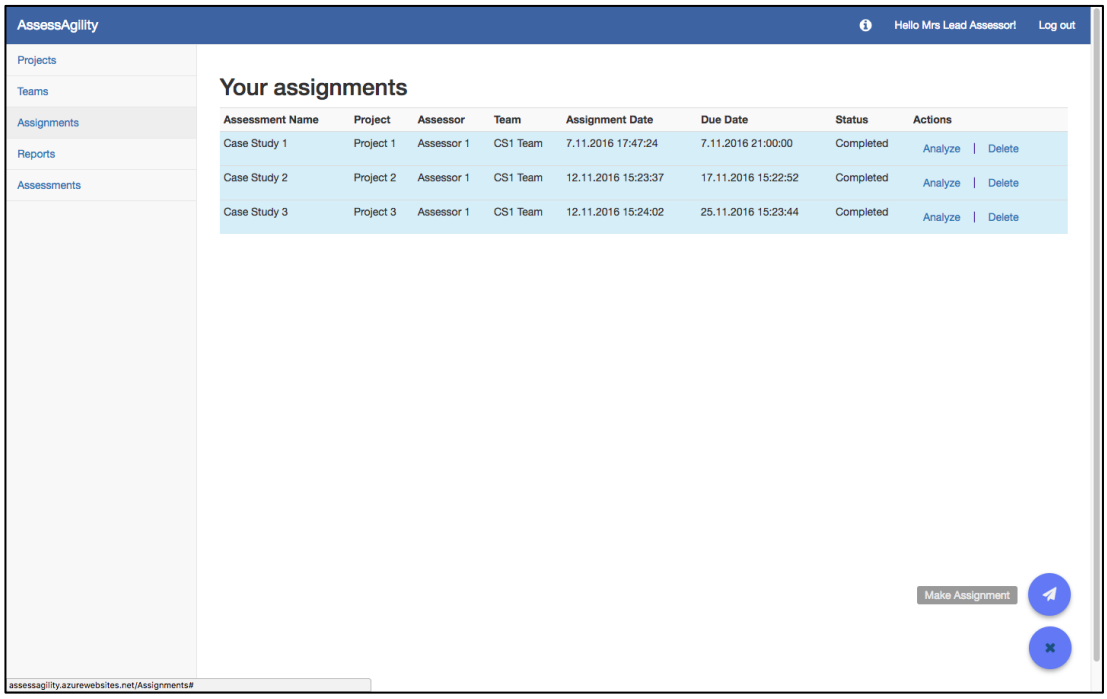


Figure 20: AssessAgility Assignments Page

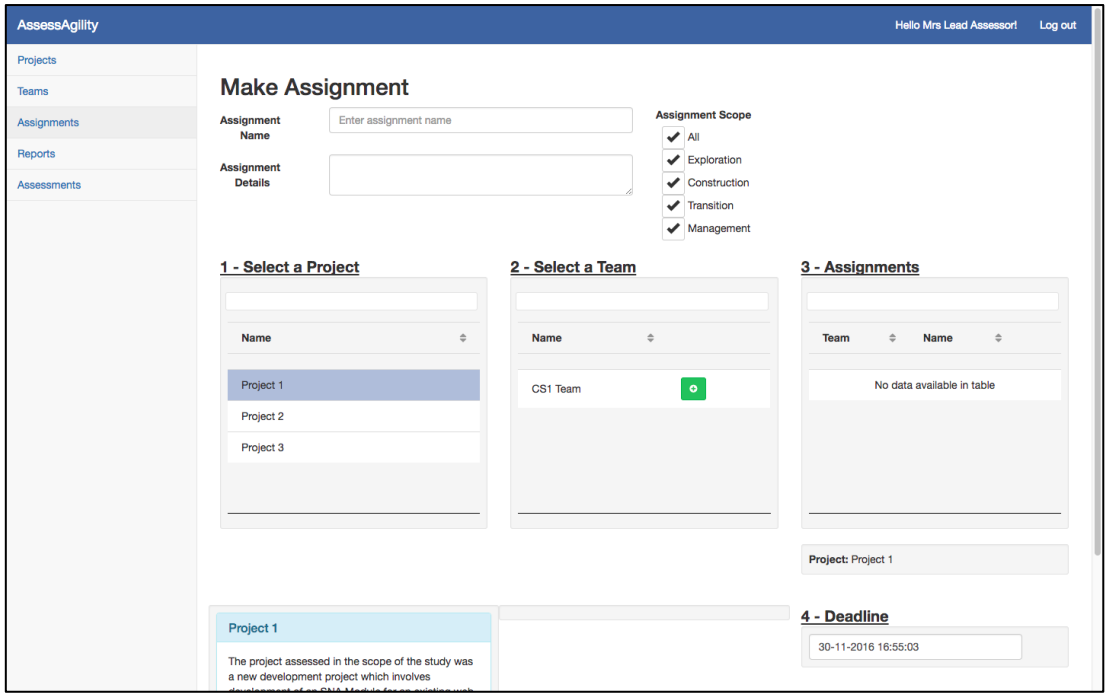


Figure 21: AssessAgility Make Assignment Page

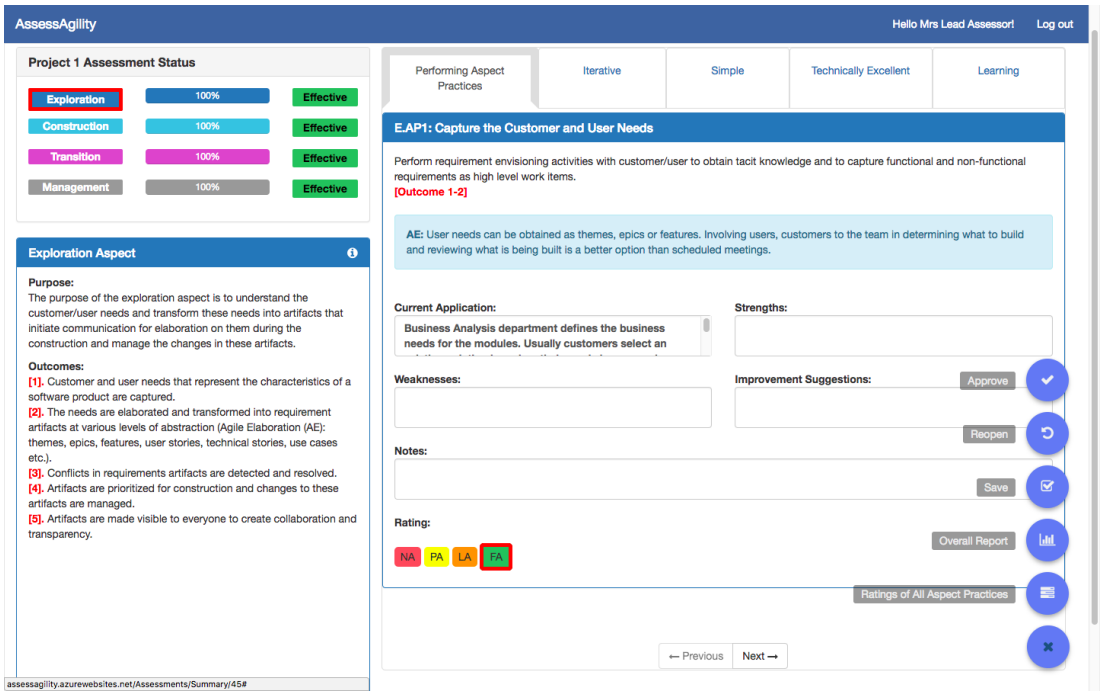


Figure 22: AssessAgility Analyze Assessment Page

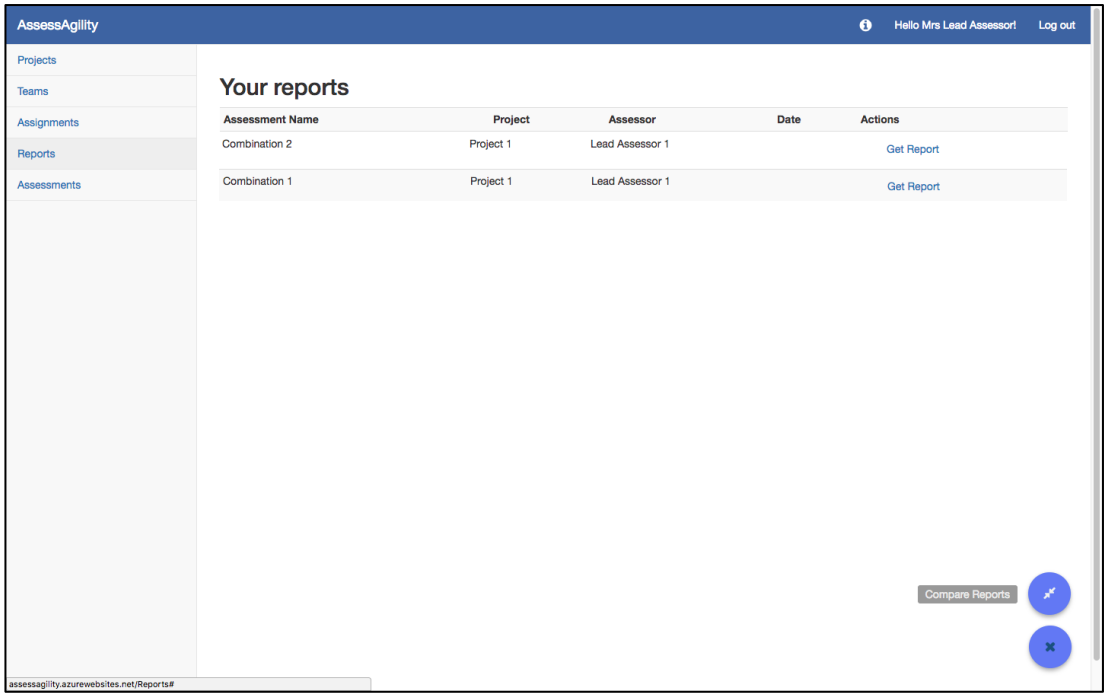


Figure 23: AssessAgility Reports Page

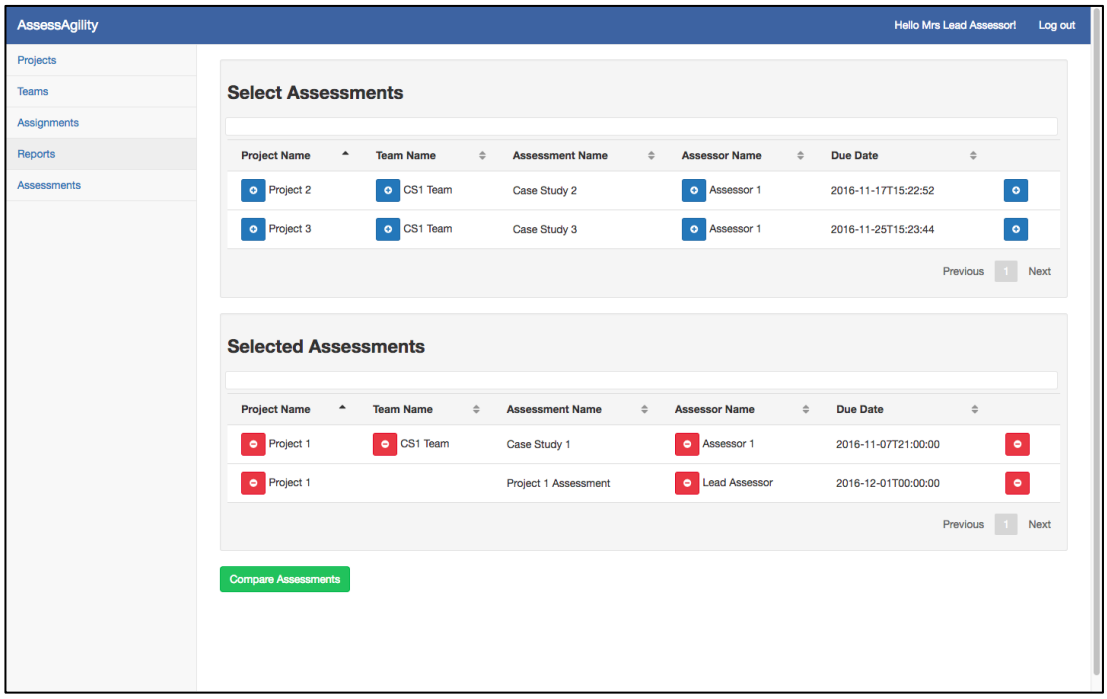


Figure 24: AssessAgility Compare Reports Page

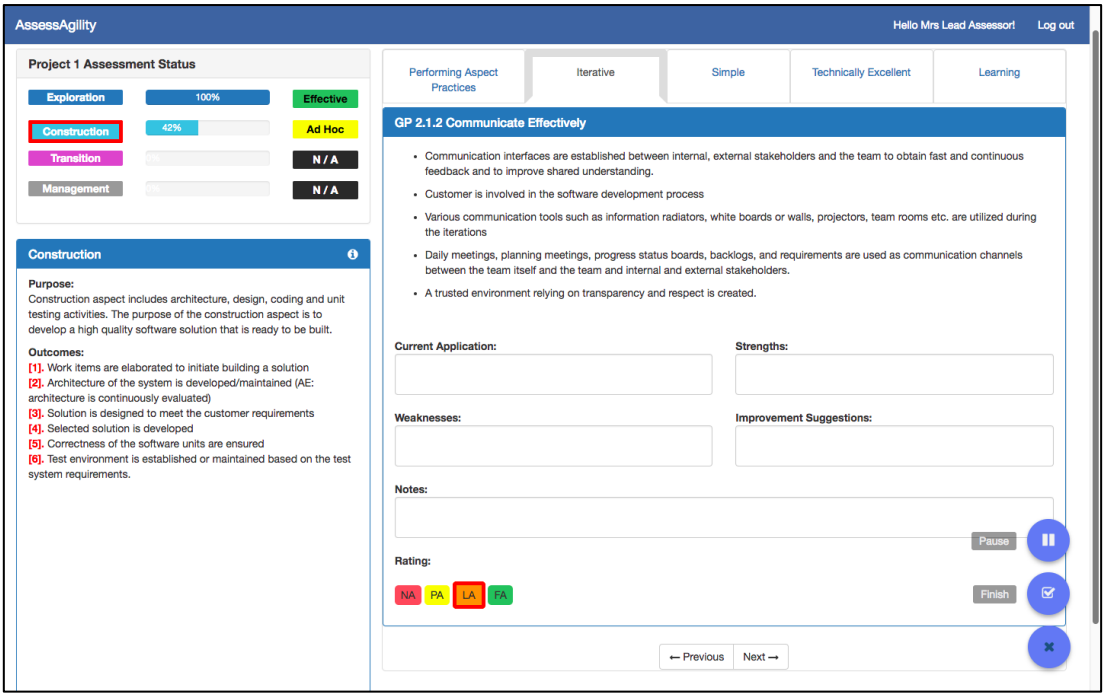


Figure 25: AssessAgility Perform Assessment Page

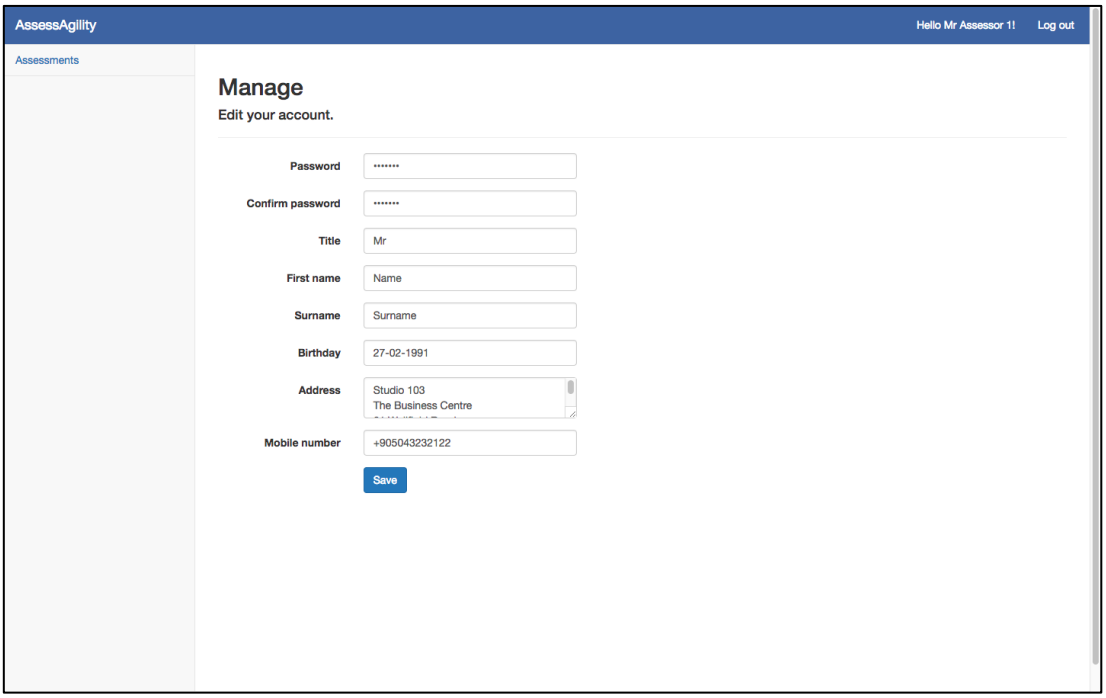


Figure 26: AssessAgility Update Profile Page

APPENDIX D

EVALUATION QUESTIONNAIRE

ID	Criteria	Question	Rating Guidance	Rating
1	Coverage	Does to the tool covers of 12 agile principles stated in the Agile Manifesto?	**Not Achieved:** 0-2 principles are covered **Partially Achieved:** 2-6 principles are covered **Largely Achieved:** 6-11 principles are covered **Fully Achieved:** 12 principles are covered	NA PA LA FA ☐ ☐ ☐ ☐
2	Availability	Does to the tool reachable via World Wide Web?	Dichotomous Scale	Not Web-Based /Web Based ☐ ☐
3	Guidance Capability	Does to the tool provide guidance for assessors who are not experts on agile software development.	**3 Possible Guidance Capabilities:** 1. providing guidance before assessment 2. providing guidance during the assessment 3. providing guidance after the assessment.	NA PA LA FA ☐ ☐ ☐ ☐

ID	Criteria	Question	Rating Guidance	Rating
			Not Achieved: None of the guidance capabilities is provided **Partially Achieved:** Only one type of guidance capability is provided **Largely Achieved:** Two types of the guidance capabilities are provided together **Fully Achieved:** All Three types of the guidance capabilities are provided together	
4	Assessment Recording	Does to the tool record agility assessment findings and provide reports for further modifications, analysis, and comparison.	Dichotomous Scale	Yes ☐ No ☐
5	Automated Reporting	Does to the tool automatically generate reports for the presentation of the results of the performed assessment.	Dichotomous Scale	Yes ☐ No ☐
6	Comparability	Does to the tool enable comparison between the reports of previously performed	Dichotomous Scale	Yes ☐ No ☐

ID	Criteria	Question	Rating Guidance	Rating
		assessments.		
7	Different Modes of Usage	Does to the tool support different usage mods for individuals, multiple users, and parallel assessments.	**3 Possible Usage Modes:** 1. single user assessment mode 2. multi-user assessment mode 3. parallel assessment mode **Not Achieved:** None of the usage modes is provided **Partially Achieved:** Only one type of usage mode is provided **Largely Achieved:** Two types of the usage modes are provided together **Fully Achieved:** All Three types of the usage modes are provided together	NA PA LA FA ☐ ☐ ☐ ☐
8	Different Scopes	Does to the tool enable performing assessments on project, team, and/or	**3 Possible Scopes:**	NA PA LA FA

ID	Criteria	Question	Rating Guidance	Rating
		organizational levels.	1. project 2. team 3. organization **Not Achieved:** None of the scope types is supported **Partially Achieved:** Only one type of scope is supported **Largely Achieved:** Two types of the scopes are supported together **Fully Achieved:** All Three types of the scopes are supported together	☐ ☐ ☐ ☐
9	Extensibility	Does to the tool provide extensibility to meet emerging needs of different types of assessment contexts.	Dichotomous Scale	Yes ☐ No ☐

APPENDIX E

USE QUESTIONNAIRE

Usefulness	**Strongly Disagree <-> Strongly Agree**	**N/A**
It helps me be more effective.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It helps me be more productive.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It is useful.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It gives me more control over the activities in the assessment process.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It makes the things I want to accomplish easier to get done.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It saves me time when I use it.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It meets my needs.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It does everything I would expect it to do.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐

Ease of Use	**Strongly Disagree <-> Strongly Agree**	**N/A**
It is easy to use.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It is simple to use.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It is user friendly.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It requires the fewest steps possible to accomplish what I want to do with it.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It is flexible.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
Using it is effortless.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
I can use it without written instructions.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
I don't notice any inconsistencies as I use it.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
Both occasional and regular users would like it.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
I can recover from mistakes quickly and easily.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
I can use it successfully every time.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐

Ease of Learning	**Strongly Disagree <-> Strongly Agree**	**N/A**
I learned to use it quickly.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
I easily remember how to use it.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It is easy to learn to use it.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
I quickly became skillful with it.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐

Satisfaction	**Strongly Disagree <-> Strongly Agree**	**N/A**
I am satisfied with it.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
I would recommend it to a friend.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It is fun to use.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It works the way I want it to work.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It is wonderful.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
I feel I need to have it.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐
It is pleasant to use.	1 2 3 4 5 6 7 ☐☐☐☐☐☐☐	☐

APPENDIX F

ASSESSMENT REPORTS FOR THE CASE STUDIES

Project 1 – Agility Assessment Report

Report Date	:	03.12.2016
Author	:	Lead Assessor
Assessor(s)	:	Assessor 1, Assessor 2
Project	:	Project 1
Project Bio	:	The project assessed in the scope of the study was a new development project which involves development of an SNA Module for an existing web service framework. It has several interfaces to other modules such as data visualization, forecasting and prediction. The framework which the developed SNA Module will be a part of is a group of web application services which provides continuous service delivery, data protection and prediction.
Organization	:	Organization GS

Ratings for Each Practice

Rating: Achievement level of a process attribute is rated based on a four point ordinal scale:

- Not Achieved (0-15% achievement percentage)
- Partially Achieved (16%-50% achievement percentage)
- Largely Achieved (51%-85% achievement percentage)
- Fully Achieved (86%-100% achievement percentage)

The chart below showcases ratings given to each practice under each aspect:

Project 1: Ratings of All Aspect Practices

1. Ad Hoc

Aspects/Practices	AP1	AP2	AP3	AP4	AP5	AP6	AP7	AP8
Exploration	FA	FA	FA	FA	FA	FA	-	-
Construction	FA	FA	FA	FA	-	-	-	-
Transition	FA	FA	FA	FA	FA	FA	-	-
Management	FA	FA	FA	FA	FA	FA	FA	FA

2. Lean

Aspects/Practices	GP 2.1.1	GP 2.1.2	GP 2.2.1	GP 2.2.2
Exploration	FA	FA	FA	FA
Construction	FA	FA	FA	FA
Transition	FA	FA	FA	FA
Management	FA	FA	FA	FA

3. Effective

Aspects/Practices	GP 3.1.1	GP 3.1.2	GP 3.2.1	GP 3.2.2	GP 3.2.3	GP 3.2.4
Exploration	FA	FA	FA	FA	FA	FA
Construction	FA	FA	FA	FA	LA	FA
Transition	FA	FA	FA	FA	LA	LA
Management	FA	FA	FA	FA	FA	FA

Achieved Agility Levels of Aspects

The bar chart below showcases achieved agility levels for each aspect.

Interpretation of Practices

Exploration Aspect

E.AP1: Capture the Customer and User Needs	
Current Application	Business Analysis department defines the business needs for the modules. Usually customers select an existing solution based on their needs however when a customer need cannot be met with the current asset base, business analysts record that customer need into Jira. The recorded customer needs are discussed in meetings with attendance of module owners and CEO. If a customer need is decided for implementation, Business Analysts turn it into a detailed business needs and add it to the Jira.
Strengths	Customer Collaboration Management of the requirements via tool Continuous communication between stakeholders
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
E.AP2: Elaborate Requirements Artifacts	
Current Application	The business needs recorded in the Jira are elaborated by the related modules' teams, business analysts and customer. These business needs are turned into detailed user stories and again recorded into Jira.
Strengths	Requirements are elaborated through collaboration and communication, User stories are managed through a tool
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
E.AP3: Detect and Resolve Conflicts of Requirements Artifacts	
Current Application	When a conflict is detected it's discussed in the sprint planning meetings and conflicts are resolved with direct communication with business analysts and customers.
Strengths	Close communication between customer and analysts, The conflict resolution progress is also recorded and kept with requirements.
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
E.AP4: Specify Dependencies Among Requirements Artifacts	
Current Application	Dependencies between customer needs, business needs and user stories are specified in Jira.
Strengths	Jira tool is used to specify dependencies amongst requirements artifacts
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
E.AP5: Manage the Requirement Artifacts	
Current Application	The user stories are placed in the backlogs according to their business value. Backlogs are adjusted with adjustment meetings that are hold weekly. When a change request comes, it is recorded but does not integrated to the current springs and discussed later. All changes are assessed in terms of risk and impact analysis.
Strengths	Usage of product backlog, Frequent and sound grooming, Sound risk and impact analysis strategies
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
E.AP6: Make the Artifacts Visible to Everyone	
Current Application	All requirement artifacts including backlog are visible to internal and external stakeholders through Jira and through organization's portal.
Strengths	Universal visibility between all parties
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.1.1 Develop Work Products in an Iterative and Incremental Way	
Current Application	All the Exploration related products are developed in an iterative and incremental way. Time-boxed iterations are employed and after each iteration a demo is hold to get customer feedback. Usually the customers of modules are the teams that are working on communicating modules. Therefore, frequent feedbacks and demos are hold between seperate teams.
Strengths	Time-boxed iterations, Frequent feedbacks and demo meetings
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
GP 2.1.2 Communicate Effectively	
Current Application	Team members get together in daily stand-up meetings. Communication channels for both internal and external stakeholders are in place. All stakeholders have access to project related artifacts (backlogs, meeting logs, issue lists) and the customer is explicitly integrated to the development process with on-site meetings.
Strengths	Daily Stand-up meetings, On-site meetings with customer
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.1 Balance the Predictive Work and Adaptive Work	
Current Application	Weekly adjustments at product backlog enable a balanced work flow by keeping the items going in and out to sprints balanced.
Strengths	Adjustments are made to the backlog to sustain the balance
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
GP 2.2.2 Employ Minimally Sufficient Ceremonies and Specify Criteria for Documentation	
Current Application	Checklists prepared from acceptance criteria recorded on the user stories, are being used to review the products. Retrospective meetings are held at the end of each sprint to gather feedback from the team. Document templates are designed with include-if-crucial mindset and these templates are used by teams. Producing unnecessary documentation or work is given a penalty according to the organization's culture.
Strengths	Usage of acceptance criteria kept within the user stories
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.1 Incorporate Agile Engineering Methods/Practices to the Aspect Practices	
Current Application	TDD is applied through acceptance criteria recorded in the user stories. Also, team members perform pair programming.
Strengths	Usage of TDD principles

Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.2 Integrate Tools to Aspects to Improve the Productivity	
Current Application	Jira is used for product and sprint backlogs. A portal is used to access to other documents.
Strengths	Shared directory usage
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.1 Support Collaborative Work and Shared Responsibility	
Current Application	Project team members collaboratively perform project related work. Decision making is done collectively. Also, all members share responsibility for Exploration artifacts except the situations where domain related knowledge is required.
Strengths	Shared rooms, Shared responsibility, Collective decision making and conflict resolution
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
GP 3.2.2 Adopt Agile Leadership Styles	
Current Application	Team Leaders do not perform traditional project management tasks. They are only representatives of the team and the team leader role is switched between team members.
Strengths	Team leaders have a vision of making the work easier for their teams
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.3 Encourage People in the Organization to Participate in Learning, Teaching and Improvement	
Current Application	Each team in the organization improves itself with retrospectives continuously. There is peer to peer learning between team members.
Strengths	P2P Learning Retrospective meetings
Weaknesses	N/A
Improvement	N/A

Suggestions	
Rating	FA
GP 3.2.4 Collect Measures to Support Learning and Improvement	
Current Application	The organization defined custom metrics to track and monitor the Exploration activities such as backlog velocity. The definitions of the metrics are kept in portal and tools are used to project the metrics.
Strengths	Goal based metric design
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

Construction Aspect

CN.AP1: Elaborate the Work Items	
Current Application	To further elaborate the user stories defined in the Exploration aspect, team creates sequence diagrams.
Strengths	Employment of Sequence Diagrams
Weaknesses	N/A
Improvement Suggestions	Team should analyze the need for the other diagrams and integrate usage of them if needed.

Rating	FA
CN.AP2: Explore the Design	
Current Application	Architects in the teams develop design solutions that are discussed and evaluated in terms of the functional and quality requirements. Class and sequence diagrams are developed to explore the designs.
Strengths	Team uses the design elements that provides direct advantages for exploring the design. Unnecessary, effort and items are eliminated.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
CN.AP3: Develop the Solution	
Current Application	For development C# and Java programming languages are used. Coding standards including comment notations are applied across teams.
Strengths	Usage of a tailored coding standard.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

CN.AP4: Ensure the Correctness of Software at Developer Level	
Current Application	Software is verified through automated unit tests. Pair reviews are employed to review the code for refactoring.
Strengths	Usage of automated unit tests and pair reviews
Weaknesses	N/A
Improvement Suggestions	Static code analysis tools can be employed.
Rating	FA
GP 2.1.1 Develop Work Products in an Iterative and Incremental Way	
Current Application	Software is developed iteratively and incrementally.
Strengths	N/A
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.1.2 Communicate Effectively	
Current Application	Team works in a shared space and communicate through daily stand-up meetings and natural communication channels.

Strengths	Daily Stand-up meetings, On-site meetings with customer
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.1 Balance the Predictive Work and Adaptive Work	
Current Application	Predictive and adaptive work for is balanced through limiting WIP and backlog adjustments. Frequent demos are made to check the solutions.
Strengths	WIP Limitations Frequent demo meetings
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.2 Employ Minimally Sufficient Ceremonies and Specify Criteria for Documentation	
Current Application	Management of design and coding activities are performed informally with peer reviews. Dependencies between design elements are stored on Jira with exported diagrams.
Strengths	Peer reviews

Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.1 Incorporate Agile Engineering Methods/Practices to the Aspect Practices	
Current Application	INVEST criteria are being used for requirements. Planning Poker technique is being employed for estimation. Automated unit and acceptance tests are applied. Peer reviews are held to validate the code against coding standard and code is refactored after the reviews.
Strengths	Appropriate usage of proved Agile practices such as (INVEST Criteria and planning poker)
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.2 Integrate Tools to Aspects to Improve the Productivity	
Current Application	SourceTree tool is used for configuration management and check out and check in mechanisms for code. SmartBear tool is used for code reviews.
Strengths	N/A

Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.1 Support Collaborative Work and Shared Responsibility	
Current Application	Team members select their tasks voluntarily and collaborate during development. Some specialty required tasks are handled by specific team members such as data analysts or visualists.
Strengths	Voluntary task selection and collaboration
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.2 Adopt Agile Leadership Styles	
Current Application	When a problem occurs, team resolves it collectively and the cause of the problem is investigated and required cautions are taken to avoid problems occurring again.
Strengths	Collective conflict resolution
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
GP 3.2.3 Encourage People in the Organization to Participate in Learning, Teaching and Improvement	
Current Application	Teams have a special directory on portal for learning. Team members share resources and record the retrospectives to that directory which is open to everyone.
Strengths	Efficient self-learning mechanisms are in place
Weaknesses	No specific training mechanism is employed
Improvement Suggestions	Trainings can be hold for specific agile related topics.
Rating	LA
GP 3.2.4 Collect Measures to Support Learning and Improvement	
Current Application	Metrics about code are kept on SmartBear tool. Defect density and review code coverage are example metrics that are collected.
Strengths	Goal based metrics are defined, Metric collection strategy is in place
Weaknesses	GQM strategy is never updated or improved after its completion.
Improvement	Metric warehouse could be analyzed and trimmed to

Suggestions	keep the metric collection updated
Rating	FA

Transition Aspect

T.AP1: Create and Manage the Development Workspace	
Current Application	All coding related artifacts are under configuration control and kept up to date. Artifacts are put into version control and check-in-check-out mechanism is used.
Strengths	Configuration Control is applied for all the artifacts in workspace, Check-in and check-out mechanisms are used
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

T.AP2: Integrate the Code	
Current Application	Continuous integration is done daily and integrated code is put into the open environment.
Strengths	CI is highly valued in the company.
Weaknesses	N/A
Improvement	N/A

Suggestions	
Rating	FA
T.AP3: Deploy the Solution	
Current Application	The build and deployment are done automatically. Automated tests are run to check the correctness of the deployed code after each deployment. Customers are given access to the deployment area.
Strengths	Deployment is automatic, Automated tests are run to check the deployments
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
T.AP4: Test the Integrated Solution	
Current Application	Test scenarios are written according to the acceptance criteria stored in the user stories. Also, non-functional attributes of the deployments are also tested. After the initial tests defects are recorded and before deployment, there's a regression and acceptance test process.
Strengths	Software is tested thoroughly against criteria defined by customer
Weaknesses	There are still manual tests for acceptance tests

Improvement Suggestions	Acceptance tests could be automatized
Rating	FA
T.AP5: Make the Progress Visible	
Current Application	Transition progress is made visible with SourceTree tool.
Strengths	Progress is visible via tools
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
T.AP6: Create the Supporting Documentation	
Current Application	Supporting documentation requirements are specified in the organization level. Specific documents are created with the goal of improving understandability of the solutions.
Strengths	Heavy documentation is eliminated
Weaknesses	N/A
Improvement Suggestions	Documentation policy could be written in a document format to guide newcomers.

Rating	FA
GP 2.1.1 Develop Work Products in an Iterative and Incremental Way	
Current Application	Transition activities are performed iteratively and incrementally.
Strengths	N/A
Weaknesses	N/A
Improvement Suggestions	Trainings can be hold for specific agile related topics.
Rating	FA
GP 2.1.2 Communicate Effectively	
Current Application	Team works in a shared space and communicate through daily stand-up meetings and natural communication channels.
Strengths	Daily Stand-up meetings, On-site meetings with customer
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.1 Balance the Predictive Work and Adaptive Work	

Current Application	Test Driven Development principle is integrated into the transition aspect. Test cases and codes are being in development with the same time as the code.
Strengths	Adoption of TDD principles
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.2 Employ Minimally Sufficient Ceremonies and Specify Criteria for Documentation	
Current Application	Not-value added activities are eliminated and decision making is being made informally.
Strengths	Ceremonies are eliminated, Document writing criteria is widely accepted by the organization
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.1 Incorporate Agile Engineering Methods/Practices to the Aspect Practices	
Current Application	TDD is adopted by the teams and automated test scenarios are developed and run.

Strengths	Employment of TDD principles
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.2 Integrate Tools to Aspects to Improve the Productivity	
Current Application	SourceTree tool is used for version control.
Strengths	N/A
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.1 Support Collaborative Work and Shared Responsibility	
Current Application	Team collaborates through all the phases.
Strengths	Self-organized and collaborative teams, Shared and voluntary responsibility between team members
Weaknesses	Some team members specialized on specific areas based on their previous experience.
Improvement	Periodic role dispersion between team members can be

Suggestions	suggested
Rating	FA
GP 3.2.2 Adopt Agile Leadership Styles	
Current Application	When a problem occurs, team resolves it collectively and the cause of the problem is investigated and required cautions are taken to avoid problems occurring again.
Strengths	Collective decision making and conflict resolution approach
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.3 Encourage People in the Organization to Participate in Learning, Teaching and Improvement	
Current Application	Teams have a special directory on portal for learning. Team members share resources and record the retrospectives to that directory which is open to everyone.
Strengths	Team members share their knowledge with each other
Weaknesses	Lack of training plan
Improvement	Trainings can be hold for specific agile related topics.

Suggestions	
Rating	LA
GP 3.2.4 Collect Measures to Support Learning and Improvement	
Current Application	Defects found are stored with the phase injection and cause information. Defect, build and deployment statistics are collected.
Strengths	Metric strategy is in place
Weaknesses	Collected metrics are not utilized to support learning and improvement.
Improvement Suggestions	Metrics could be analyzed to support learning and improvement objectives.
Rating	LA

Management Aspect

M.AP1: Initiate the Project	
Current Application	Each module in the framework is decided and developed according to the feasibility studies made. 2 page module bios are prepared that include vision and scope of the projects.
Strengths	Feasibility Studies are conducted before the initiation of module development projects, Non-heavy but sufficient

	documentation is used
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
M.AP2: Form the Team	
Current Application	Team members are allocated to teams according to their experiences and expertize.
Strengths	N/A
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
M.AP3: Align with Agile Values and Principles	
Current Application	Team members educate themselves regularly on Agile values and practices. External stakeholders are aligned with Agile approach of the organization.
Strengths	Agile values and principles are communicated internally and externally
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
M.AP4: Establish the Physical Work Space	
Current Application	Offices have open and private spaces to facilitate communication and privacy.
Strengths	Offices provide private and communal space for the teams
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
M.AP5: Plan the Progress	
Current Application	Historical data is used to make plans within the early sprint planning. The previous backlogs are used as proxies to estimate the effort and time (in terms of sprints) needed to develop new modules. Plans are structured according to the business value so high-value items are prioritized.
Strengths	Teams keep and update their historical data for planning Estimates are also based on the historical proxies
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
M.AP6: Estimate the Work Items	
Current Application	Teams utilize historical data and proxy based estimation (PROBE) to make estimates.
Strengths	Historical data and proxies are used for estimation
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
M.AP7: Monitor the Progress	
Current Application	The progress of projects is tracked internally by team leaders and externally by the process team. The results of the monitoring are shared with teams.
Strengths	Internal processes of the organization are tracked by process engineers through metrics
Weaknesses	N/A
Improvement Suggestions	N/A

Rating	FA
M.AP8: Manage and Mitigate the Risks	
Current Application	Project risks are generated by the module owners. Then the risks are approved, prioritized and tracked. Risk mitigation strategies are discussed with all stakeholders and corrective actions are taken.
Strengths	A sound risk management strategy is in place
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.1.1 Develop Work Products in an Iterative and Incremental Way	
Current Application	Monitoring and tracking activities are handled iteratively and incrementally.
Strengths	N/A
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.1.2 Communicate Effectively	

Current Application	Communication is effective.
Strengths	Daily Stand-up meetings, On-site meetings with customer
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.1 Balance the Predictive Work and Adaptive Work	
Current Application	Plans and estimations are made and tailored continuously.
Strengths	Frequently adjusted plans, Historic data based plans
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.2 Employ Minimally Sufficient Ceremonies and Specify Criteria for Documentation	
Current Application	Management activities are done informally without supervision. Teams are self-organizing.
Strengths	Self-organizing teams, Informal management

Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.1 Incorporate Agile Engineering Methods/Practices to the Aspect Practices	
Current Application	Plans and estimations are made continuously and updated as more information is obtained through the progress.
Strengths	N/A
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.2 Integrate Tools to Aspects to Improve the Productivity	
Current Application	Jira tool is used for management related aspect practices.
Strengths	N/A
Weaknesses	N/A
Improvement	N/A

Suggestions	
Rating	FA
GP 3.2.1 Support Collaborative Work and Shared Responsibility	
Current Application	Plans and estimates are made collaboratively by the team members.
Strengths	Self-organized and collaborative teams, Shared and voluntary responsibility between team members
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.2 Adopt Agile Leadership Styles	
Current Application	Team solves its own problems by quickly investigating and developing solutions.
Strengths	In point and collaborative problem solution, Root cause analysis
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

GP 3.2.3 Encourage People in the Organization to Participate in Learning, Teaching and Improvement	
Current Application	Teams have a special directory on portal for learning. Team members share resources and record the retrospectives to that directory which is open to everyone.
Strengths	Learning is promoted throughout the organization
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

GP 3.2.4 Collect Measures to Support Learning and Improvement	
Current Application	The variance measures are collected for the estimations and actual values to learn the estimation mistakes and improve the planning accuracy.
Strengths	Actual and planned values are tracked and variances are investigated
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

Project 2 – Agility Assessment Report

Report Date	:	03.12.2016
Author	:	Lead Assessor
Assessor(s)	:	Assessor 1, Assessor 2
Project	:	Project 2
Project Bio	:	The project assessed in the scope of the study was an existing solution implementation case which involved usage of an existing platform. It involved implementation of remote device management and communication services to a manufacturing factory.
Organization	:	Organization TX

Ratings for Each Practice

Rating: Achievement level of a process attribute is rated based on a four point ordinal scale:

- Not Achieved (0-15% achievement percentage)
- Partially Achieved (16%-50% achievement percentage)
- Largely Achieved (51%-85% achievement percentage)
- Fully Achieved (86%-100% achievement percentage)

The chart below showcases ratings given to each practice under each aspect:

Project 2: Ratings of All Aspect Practices

1. Ad Hoc

Aspects/Practices	AP1	AP2	AP3	AP4	AP5	AP6	AP7	AP8
Exploration	FA	FA	FA	FA	FA	FA	-	-
Construction	FA	LA	LA	NA	-	-	-	-
Transition	FA	LA	FA	LA	FA	FA	-	-
Management	NA	FA	PA	FA	FA	NA	FA	NA

2. Lean

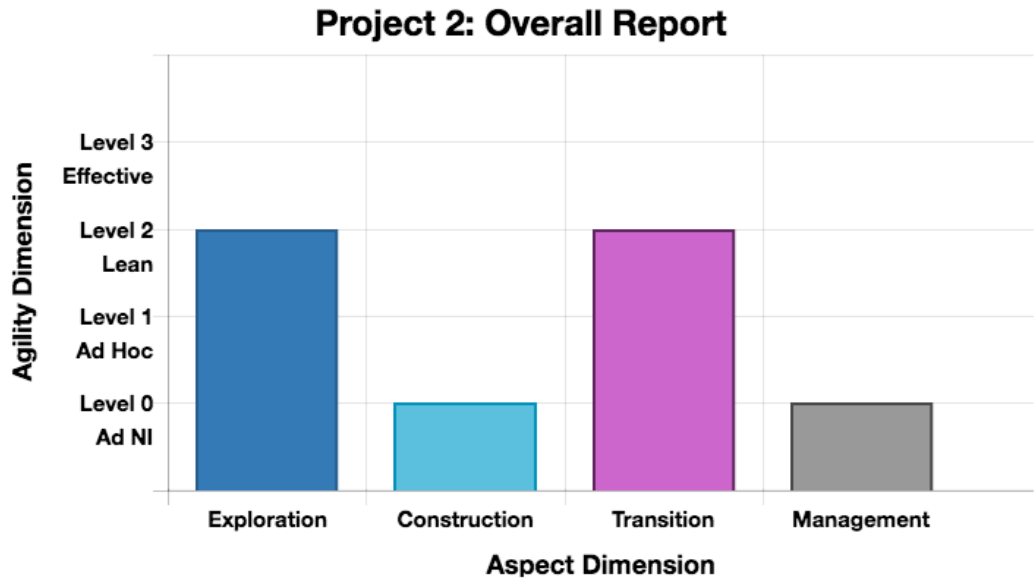
Aspects/Practices	GP 2.1.1	GP 2.1.2	GP 2.2.1	GP 2.2.2
Exploration	FA	FA	FA	FA
Construction	FA	FA	FA	PA
Transition	FA	FA	FA	FA
Management	NA	FA	NA	LA

3. Effective

Aspects/Practices	GP 3.1.1	GP 3.1.2	GP 3.2.1	GP 3.2.2	GP 3.2.3	GP 3.2.4
Exploration	FA	FA	FA	PA	PA	PA
Construction	PA	FA	FA	PA	PA	NA
Transition	PA	FA	FA	PA	PA	FA
Management	NA	FA	PA	PA	PA	NA

Achieved Agility Levels of Aspects

The bar chart below showcases achieved agility levels for each aspect.



Interpretation of Practices

Exploration Aspect

E.AP1: Capture the Customer and User Needs	
Current Application	The mode of operations of the provided solutions to customers are kept as "usage scenarios" and customers select the best fitting scenario for their needs.
Strengths	Detailed Usage Scenarios are used to capture the requirements
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
E.AP2: Elaborate Requirements Artifacts	

Current Application	Existing "usage scenarios" consists of set of use cases. These sets are discussed with customer to tailor them to their own needs. Then the steps of the use cases are altered or new steps are added or existing ones are moved. These alterations are made in direct meetings with customers.
Strengths	Usage Scenarios are elaborated with customers
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
E.AP3: Detect and Resolve Conflicts of Requirements Artifacts	
Current Application	Usage Scenario selection and use case tailoring are done with direct involvement of the customers so conflicts are resolved with customers involvement.
Strengths	Customer involvement to conflict resolution
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
E.AP4: Specify Dependencies Among Requirements Artifacts	
Current Application	Selected usage stories and tailored use cases are kept in Wrike Tool. The dependencies of the usage stories and

	use cases are established within the tool.
Strengths	Effective usage of the tool
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
E.AP5: Manage the Requirement Artifacts	
Current Application	The product backlog is created and managed through the Wrike Tool. The changes to the items are tracked and necessary adjustments (i.e. re-priotizing) are made to the backlog as changes appear.
Strengths	Usage of Product Backlog, Value based prioritization, Effective tool usage
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
E.AP6: Make the Artifacts Visible to Everyone	
Current Application	All the requirement artifacts are visible to both customer and team members through Wrike. Sometimes role based access control feature of the tool is utilized to separate technical items from generic items.

Strengths	High visibility, Role-based access control to avoid over sharing
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.1.1 Develop Work Products in an Iterative and Incremental Way	
Current Application	Requirement artifacts are developed in an iterative and incremental way.
Strengths	The strengths are captured in Current Application.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.1.2 Communicate Effectively	
Current Application	Team works in a shared space and communication between team members are supported with daily stand up meetings. Also, the communication feature of Wrike (@mention) is used to communicate anytime anywhere within the team and the customer.
Strengths	N/A

Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.1 Balance the Predictive Work and Adaptive Work	
Current Application	The flow of the work is balanced through regular cycle planning gatherings between team members. Workload is balanced.
Strengths	The strengths are captured in Current Application.
Weaknesses	Team makes up-front decisions and sticks to them.
Improvement Suggestions	Decisions should be mad as late as possible.
Rating	FA
GP 2.2.2 Employ Minimally Sufficient Ceremonies and Specify Criteria for Documentation	
Current Application	Requirements are approved informally. The wrike tool is used for informal decision making with the involvement of the customer.
Strengths	There's a policy about not writing unnecessary documents.
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
GP 3.1.1 Incorporate Agile Engineering Methods/Practices to the Aspect Practices	
Current Application	Backlog approach is used to keep requirement items in prioritized order.
Strengths	Usage of Product Backlog
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.2 Integrate Tools to Aspects to Improve the Productivity	
Current Application	Wrike tool is used for requirements management.
Strengths	Effective tool usage.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

GP 3.2.1 Support Collaborative Work and Shared Responsibility	
Current Application	Team members share responsibility of the requirements all together. They own the requirements and manage them collaboratively.
Strengths	Joint ownership of requirements
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.2 Adopt Agile Leadership Styles	
Current Application	The team members work in a unison without a command and control approach however there's still a project manager role which manages teams with command and control style approach.
Strengths	Team based Agile Leadership
Weaknesses	Project Manager employs command and control
Improvement Suggestions	Project Managers could be trained on Agile Leadership Styles and their benefits.
Rating	PA
GP 3.2.3 Encourage People in the Organization to Participate in Learning, Teaching and Improvement	

Current Application	Knowledge is shared between team members and with mentoring approach.
Strengths	Mentoring
Weaknesses	There's no agile specific learning approach visible in the organization.
Improvement Suggestions	Agile focused approaches could be set in place to build an organization culture.
Rating	PA
GP 3.2.4 Collect Measures to Support Learning and Improvement	
Current Application	Measures provided by the Wrike tool are collected however not used.
Strengths	Strengths are captured in Current Application.
Weaknesses	Observation is used instead of metrics.
Improvement Suggestions	A strategy about establishing how to interpret the collected measures is needed.
Rating	PA

Construction Aspect

CN.AP1: Elaborate the Work Items	
Current Application	Use cases located in the backlog are elaborated by the team member working on it. Just in time detailing is

	done.
Strengths	Just in time elaboration
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
CN.AP2: Explore the Design	
Current Application	Each team member develops the design about the backlog item he/she is working on.
Strengths	Strengths are captured in Current Application
Weaknesses	Lack of design discussion
Improvement Suggestions	Designs can be communicated between team members to discuss alternative solutions and approaches.
Rating	LA
CN.AP3: Develop the Solution	
Current Application	Each use case is developed by the developers. However, the understandability of the code is reduced because of lack of comments.
Strengths	Strengths are captured in Current Application

Weaknesses	There's no coding standard or commenting mechanism.
Improvement Suggestions	A coding standard can be established.
Rating	LA
CN.AP4: Ensure the Correctness of Software at Developer Level	
Current Application	Developer level tests are not consistent. Some developers perform automated tests and some are not.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application
Improvement Suggestions	Automated tests can be applied.
Rating	NA
GP 2.1.1 Develop Work Products in an Iterative and Incremental Way	
Current Application	Software is developed in an iterative and incremental way. Backlog is used and frequent demos are made to the customer.
Strengths	Backlog usage, Iterative and incremental development
Weaknesses	N/A
Improvement Suggestions	N/A

Rating	FA
GP 2.1.2 Communicate Effectively	
Current Application	Development Team shares the same room and frequent customer visits are made to the development team to discuss changes and for demos. Wrike tool is again used for distributed communication between team members and customer.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.1 Balance the Predictive Work and Adaptive Work	
Current Application	The flow of the work is balanced through regular cycle planning gatherings between team members. Workload is balanced.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

GP 2.2.2 Employ Minimally Sufficient Ceremonies and Specify Criteria for Documentation	
Current Application	Retrospectives are held for identification of non value added activities. However, ceremonies are made with the project manager.
Strengths	Time limited Retrospectives
Weaknesses	Ceremonies at project manager level
Improvement Suggestions	Project management approach and project manager role should be aligned with the Agile approach which is established in team level.
Rating	PA
GP 3.1.1 Incorporate Agile Engineering Methods/Practices to the Aspect Practices	
Current Application	Pair programming is done.
Strengths	Strengths are captured in Current Application
Weaknesses	No specific approach to ensure correctness of the software.
Improvement Suggestions	TDD approach can be adopted to ensure the correctness of the software.
Rating	PA
GP 3.1.2 Integrate Tools to Aspects to Improve the Productivity	

Current Application	GIT tool and committing mechanism used for version and change control.
Strengths	Tool usage
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.1 Support Collaborative Work and Shared Responsibility	
Current Application	Team members select tasks on their own, the responsibility of the code is shared between members. The parts of the software that requires interaction between each other are developed collaboratively.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.2 Adopt Agile Leadership Styles	
Current Application	Team features a champion that shields the development process from the upper project manager. However, there's still command and control approach on the project assignment level.

Strengths	Internal task assignment mechanism of the team is voluntarily.
Weaknesses	Top level project manager.
Improvement Suggestions	Project manager could be trained on Agile Methodologies.
Rating	PA
GP 3.2.3 Encourage People in the Organization to Participate in Learning, Teaching and Improvement	
Current Application	Learning is encouraged within the team. Mentoring and pair programming are used as learning mechanisms.
Strengths	Mentoring, Pair Programming
Weaknesses	No organizational learning objectives are in place about agile practices.
Improvement Suggestions	Organizational learning objectives could be set.
Rating	PA
GP 3.2.4 Collect Measures to Support Learning and Improvement	
Current Application	Construction related measures are not collected.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application

Improvement Suggestions	N/A
Rating	NA

Transition Aspect

T.AP1: Create and Manage the Development Workspace	
Current Application	GIT is used for configuration control. Changes made to the artifacts are made with check-in and check-out mechanisms.
Strengths	Tool usage
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
T.AP2: Integrate the Code	
Current Application	Integration is performed automatically with scripts. System can be built with a single command.
Strengths	Strengths are captured in Current Application
Weaknesses	Tests are not automated.
Improvement Suggestions	Automated tests can be utilized after integrations to ensure rapid feedback.

Rating	LA
T.AP3: Deploy the Solution	
Current Application	Deployment is performed continuously and automatically. There're seperate environments for development and deployment. Each deployed build is tested by the testers.
Strengths	Continuous and automatic deployment, Different environments for development and deployment
Weaknesses	No automated tests
Improvement Suggestions	Automated tests can be utilized to ensure correctness of the deployed builds.
Rating	FA
T.AP4: Test the Integrated Solution	
Current Application	Regression, integration and acceptance tests are run on the deployed builds by the testers. Acceptance criteria on usage stories are used for the test cases.
Strengths	Strengths are captured in Current Application
Weaknesses	No automated tests
Improvement Suggestions	Automated tests can be utilized to ensure correctness.
Rating	LA

T.AP5: Make the Progress Visible	
Current Application	Transition progress is visible to internal and external stakeholders.
Strengths	Internal and external visibility is ensured.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
T.AP6: Create the Supporting Documentation	
Current Application	Build and deployment statuses are viewed by the customer on the GIT. Also, test results, defects found and solutions are published on the Wrike tool for everyone to see, edit and make comments.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.1.1 Develop Work Products in an Iterative and Incremental Way	
Current Application	Transition aspect activities are performed in multiple iterations. The deployments are planned within backlog

	planning. Frequent demos are made to the customer.
Strengths	Employment of multiple iterations, Frequent demos
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.1.2 Communicate Effectively	
Current Application	Team works in a shared space and communication between team members are supported with daily stand up meetings. Also the communication feature of Wrike (@mention) is used to communicate anytime anywhere within the team and the customer.
Strengths	Shared workspace, Direct communication via tool
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.1 Balance the Predictive Work and Adaptive Work	
Current Application	The flow of the transition aspect related work is balanced through regular cycle planning gatherings between team members. Workload is balanced.

Strengths	Cycle Planning Meetings
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.2 Employ Minimally Sufficient Ceremonies and Specify Criteria for Documentation	
Current Application	No additional ceremonies are held except demonstrations with customers. No additional meetings are held.
Strengths	Document writing standard is established through organization and criteria is designed to avoid non-value added parts.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.1 Incorporate Agile Engineering Methods/Practices to the Aspect Practices	
Current Application	Continuous integration is adopted but manual tests are employed.
Strengths	Continuous Integration Strategy

Weaknesses	Manual Tests
Improvement Suggestions	Automated test suites can be utilized to test the deployments.
Rating	PA
GP 3.1.2 Integrate Tools to Aspects to Improve the Productivity	
Current Application	GIT and Wrike tools are used through transition.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.1 Support Collaborative Work and Shared Responsibility	
Current Application	The responsibility of the transition activities is shared between members.
Strengths	Shared responsibility
Weaknesses	N/A
Improvement Suggestions	N/A

Rating	FA
GP 3.2.2 Adopt Agile Leadership Styles	
Current Application	Team features a champion that shields the development process from the upper project manager. However, there's still command and control approach on the project by the manager.
Strengths	Strengths are captured in Current Application
Weaknesses	Top level project manager.
Improvement Suggestions	Project manager could be trained on Agile Methodologies.
Rating	PA
GP 3.2.3 Encourage People in the Organization to Participate in Learning, Teaching and Improvement	
Current Application	Learning is encouraged within the team. Mentoring and pair programming are used as learning mechanisms.
Strengths	Mentoring, Pair Programming
Weaknesses	No organizational learning objectives are in place about agile practices for transition.
Improvement Suggestions	An organizational learning plan could be created.
Rating	PA

GP 3.2.4 Collect Measures to Support Learning and Improvement	
Current Application	Check-in and check-out numbers and deployment frequency are tracked.
Strengths	The collected measures are analyzed regularly to track and improve the delivery frequency to customer.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

Management Aspect

M.AP1: Initiate the Project	
Current Application	Feasibility studies are held at the beginning of the organization. Projects do not include separate feasibility studies or vision. Scope is defined with the selected usage stories.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application
Improvement Suggestions	Initiation strategies including feasibility studies could be performed for individual projects
Rating	NA

M.AP2: Form the Team	
Current Application	Team is formed according to the experience and knowledge.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
M.AP3: Align with Agile Values and Principles	
Current Application	Customers are aware of the fact that agile is applied through development. However, there's an inconsistency between project manager and the team.
Strengths	External stakeholders are aligned
Weaknesses	Project Manager insists on traditional project management values and principles
Improvement Suggestions	Project manager could be trained on Agile Methodologies.
Rating	PA
M.AP4: Establish the Physical Work Space	
Current Application	Physical workspace is appropriate for agile.

Strengths	Shared space, Quite and solitary rooms are available for team members
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
M.AP5: Plan the Progress	
Current Application	The development plan is prepared and maintained iteratively. Daily activities are coordinated through daily stand up meetings.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
M.AP6: Estimate the Work Items	
Current Application	Estimation is done via expert judgment.
Strengths	N/A
Weaknesses	Sole reliance on expert judgment

Improvement Suggestions	A proved estimation methodology could be adopted and historical database could be set for future estimations.
Rating	NA
M.AP7: Monitor the Progress	
Current Application	The progress of the team is monitored by the project manager with the Wrike tool. Effort, schedule and cost are monitored and updated through tool's dashboards.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
M.AP8: Manage and Mitigate the Risks	
Current Application	Project risks are not tracked.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application
Improvement Suggestions	A risk management strategy that is aligned with organizational objectives could be developed and used.
Rating	NA

GP 2.1.1 Develop Work Products in an Iterative and Incremental Way	
Current Application	Management related work products are not developed in an iterative and incremental way.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application
Improvement Suggestions	Project management plans and estimations could be developed iteratively and incrementally as more information obtained through development cycle.
Rating	NA
GP 2.1.2 Communicate Effectively	
Current Application	Project Manager and team member communicates effectively through weekly meetings. Customer is involved in the meetings.
Strengths	Weekly and customer involved meetings
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.1 Balance the Predictive Work and Adaptive Work	
Current Application	Project management related decisions are not taken

	collaboratively.
Strengths	N/A
Weaknesses	Project manager has absolute responsibility over management decisions.
Improvement Suggestions	Team members should be given opportunity to have their say for management related work.
Rating	NA
GP 2.2.2 Employ Minimally Sufficient Ceremonies and Specify Criteria for Documentation	
Current Application	Informal procedures are applied to handle management decisions.
Strengths	Strengths are captured in Current Application
Weaknesses	Retrospectives are only done at the team level.
Improvement Suggestions	Project level retrospectives could be employed.
Rating	LA
GP 3.1.1 Incorporate Agile Engineering Methods/Practices to the Aspect Practices	
Current Application	Agile methods are not incorporated to management work.

Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application
Improvement Suggestions	Effort estimation could be based on historical data or function point estimation.
Rating	NA
GP 3.1.2 Integrate Tools to Aspects to Improve the Productivity	
Current Application	Wrike tool is utilized for management aspect activities.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.1 Support Collaborative Work and Shared Responsibility	
Current Application	Collaboration between project manager and team champion is visible. There's an underlying estimation collaboration between team members and champion.
Strengths	Planning and estimation processes.
Weaknesses	Team is not directly involved in planning and estimation processes.

Improvement Suggestions	Team could be involved in planning and estimation processes.
Rating	PA
GP 3.2.2 Adopt Agile Leadership Styles	
Current Application	Project Manager still employs commanding and controlling style of management on the team. However, teams have leaders that enables people to work without command and control mechanism.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application
Improvement Suggestions	Project manager could be trained on Agile Methodologies.
Rating	PA
GP 3.2.3 Encourage People in the Organization to Participate in Learning, Teaching and Improvement	
Current Application	Learning is encouraged within the team. Mentoring and pair programming are used as learning mechanisms.
Strengths	Strengths are captured in Current Application
Weaknesses	No organizational learning objectives are in place about agile practices for construction.
Improvement Suggestions	An organizational learning plan could be created.

Rating	PA
GP 3.2.4 Collect Measures to Support Learning and Improvement	
Current Application	Management aspect's activities are not followed through measures.
Strengths	N/A
Weaknesses	Management activities are not monitored and tracked.
Improvement Suggestions	Managers could track and monitor their work and discuss the findings with teams and each other.
Rating	NA

Project 3 – Agility Assessment Report

Report Date	:	03.12.2016
Author	:	Lead Assessor
Assessor(s)	:	Assessor 1, Lead Assessor 2
Project	:	Project 3

Project Bio	:	The project assessed in the scope of the study was a new development project that involves development of a third person action adventure game. The game is based on the neural network AI technologies and has the ability to adapt the main scenario to player's gaming style. The project was developed with the Unity Game Engine.
Organization Bio	:	Organization BV

Ratings for Each Practice

Rating: Achievement level of a process attribute is rated based on a four point ordinal scale:

• Not Achieved (0-15% achievement percentage)
• Partially Achieved (16%-50% achievement percentage)
• Largely Achieved (51%-85% achievement percentage)
• Fully Achieved (86%-100% achievement percentage)

The chart below showcases ratings given to each practice under each aspect:

Project 3: Ratings of All Aspect Practices

1. Ad Hoc

Aspects/Practices	AP1	AP2	AP3	AP4	AP5	AP6	AP7	AP8
Exploration	FA	FA	FA	FA	FA	FA	-	-
Construction	FA	FA	FA	PA	-	-	-	-
Transition	PA	PA	NA	NA	LA	PA	-	-
Management	FA	FA	FA	FA	LA	LA	FA	NA

2. Lean

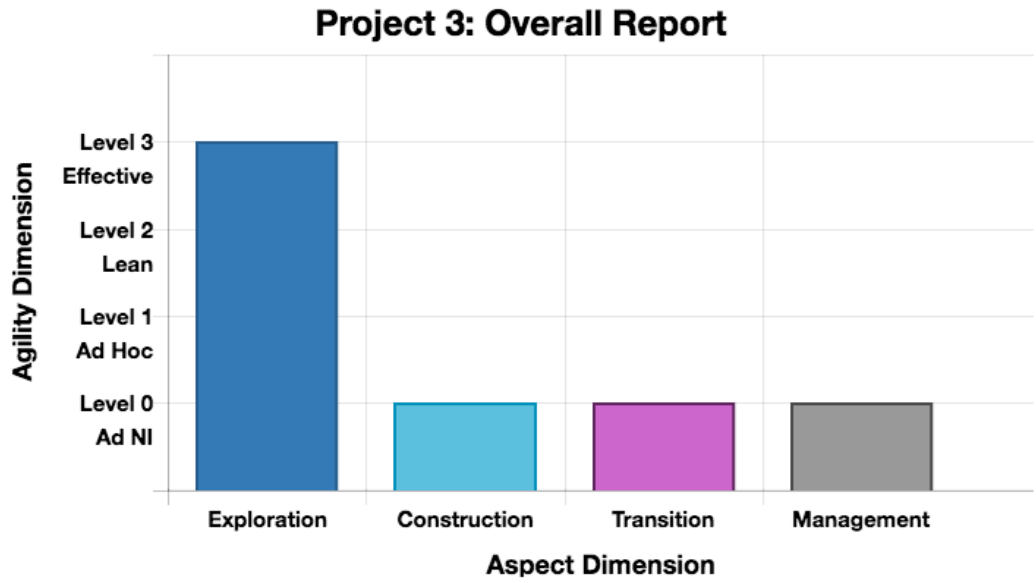
Aspects/Practices	GP 2.1.1	GP 2.1.2	GP 2.2.1	GP 2.2.2
Exploration	FA	FA	FA	FA
Construction	FA	FA	FA	FA
Transition	LA	LA	PA	LA
Management	FA	FA	NA	FA

3. Effective

Aspects/Practices	GP 3.1.1	GP 3.1.2	GP 3.2.1	GP 3.2.2	GP 3.2.3	GP 3.2.4
Exploration	FA	FA	FA	FA	FA	FA
Construction	PA	FA	FA	FA	FA	NA
Transition	NA	PA	PA	LA	LA	NA
Management	PA	FA	FA	FA	LA	FA

Achieved Agility Levels of Aspects

The bar chart below showcases achieved agility levels for each aspect.



Interpretation of Practices

Exploration Aspect

E.AP1: Capture the Customer and User Needs	
Current Application	The company has an idea board, staff writes their game ideas to the board and then ideas are voted regularly. Top 3 is picked for the development. Then the owner of each idea writes the high-level requirements for each game.
Strengths	Requirements are captured as high level work items
Weaknesses	N/A
Improvement Suggestions	N/A

Rating	FA
E.AP2: Elaborate Requirements Artifacts	
Current Application	After the owner writes the high-level requirements, Game Design Documents are prepared including the story, game item, objectives and dependencies. Then rough sketches for the screens are designed. Each sketch is designed addressing specific requirements (taken as side notes with the screens) and exported and stored on the Mural tool. Mural enables the connection between high level requirements and sketches.
Strengths	Tool usage for keeping high level work items and detailed work items
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
E.AP3: Detect and Resolve Conflicts of Requirements Artifacts	
Current Application	Conflicts related to the requirements are resolved with team members discussing with each other on white board.
Strengths	Collaborative conflict resolution, Visualization of conflicts
Weaknesses	N/A
Improvement	N/A

Suggestions	
Rating	FA
E.AP4: Specify Dependencies Among Requirements Artifacts	
Current Application	Mural tool is used to specify and keep the dependencies between requirement artifacts. High level requirements, screen designs, graphic designs and notes are related to each other.
Strengths	Dependencies are kept on a tool, Artifacts are linked to each other
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
E.AP5: Manage the Requirement Artifacts	
Current Application	A Kanban board is kept to manage and prioritize the work items. Changes are discussed collaboratively and re-priotizing is done when a change occurs.
Strengths	Kanban methodology is effectively employed to manage artifacts
Weaknesses	N/A
Improvement Suggestions	N/A

Rating	FA
E.AP6: Make the Artifacts Visible to Everyone	
Current Application	Requirements artifacts are made visible to everyone by the Mural tool.
Strengths	Visibility is obtained through tools
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.1.1 Develop Work Products in an Iterative and Incremental Way	
Current Application	Requirements artifacts are developed in an iterative and incremental way. Screen designs enable visual demonstration from early phases and detection of misunderstandings.
Strengths	Iterative and incremental development is adopted and used Early feedback is obtained through visualized artifacts
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

GP 2.1.2 Communicate Effectively	
Current Application	Team shares the same room and daily stand-up meetings (or as the team says grab-a-coffee) meetings are held to communicate and discuss daily activities and problems.
Strengths	Shared room, Stand-up meetings
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.1 Balance the Predictive Work and Adaptive Work	
Current Application	Work is balanced through Kanban board. Team commits to limiting WIP. Team plans cycles by choosing the work items that will be developed in that cycle.
Strengths	Kanban Methodology is used effectively to limit WIP Cycle planning
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.2 Employ Minimally Sufficient Ceremonies and Specify Criteria for Documentation	

Current Application	The team has a policy that they're calling just-make-it-happen. The policy states that no ceremony should be held unless it's necessary. Also team members state that they're applying RUP rules to the documents that is: "If document does not increases your understanding of the product then it's unnecessary"
Strengths	Ceremony is internally minimized throughout the team, A minimal document writing criteria is in place
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.1 Incorporate Agile Engineering Methods/Practices to the Aspect Practices	
Current Application	Kanban practices such as visualizing the work, limiting WIP, managing flow, explicit process and feedback loops are applied.
Strengths	Kanban, Frequent Feedback Loops, Limited WIP, Managed Flow
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.2 Integrate Tools to Aspects to Improve the Productivity	

Current Application	Mural tool is used for requirements and project management.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.1 Support Collaborative Work and Shared Responsibility	
Current Application	Team collaborates as a whole and it's self-organizing. Team members are sharing the responsibility of the game however they're organized according to their interests and expertize.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.2 Adopt Agile Leadership Styles	
Current Application	Game owner leads the team with agile leadership styles and focuses on guiding people to do their works. Mistakes and problems are discussed without blaming and each mistake is embraced as a lesson learned.

Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.3 Encourage People in the Organization to Participate in Learning, Teaching and Improvement	
Current Application	Mural is used for team members pointing the new technology that they should learn and apply. Also, teams constantly attend to educational events and conferences on the subjects such as UX Design and Agile Methodologies.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.4 Collect Measures to Support Learning and Improvement	
Current Application	Screen number is used as a measure to track the requirements. Then the team categorizes the screens according to a complexity scheme they have developed from previous experiences. This metric is collected and analyzed for size and effort estimates.

Strengths	Custom measures designed for the needs of the team are in place
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

Construction Aspect

CN.AP1: Elaborate the Work Items	
Current Application	In the exploration activities, the requirements are visualized with rough sketches. In construction, each sketch is turned into detailed screens with specific elements and solutions (taken as side notes with the screens) then exported and stored on the Mural tool. Mural enables the connection between high level requirements, sketches and detailed screens.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
CN.AP2: Explore the Design	
Current Application	User Interface Prototyping is used as the default design exploration technique. Also, some teams are trying to

	adapt UX design first approach.
Strengths	UI Prototyping, UX Design
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
CN.AP3: Develop the Solution	
Current Application	Software is developed with Unity Game Engine. This enables the team to develop platform independent games.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
CN.AP4: Ensure the Correctness of Software at Developer Level	
Current Application	Software is tested manually on the user interfaces.
Strengths	N/A

Weaknesses	There's no coding standard.
Improvement Suggestions	Peer reviews, pair programming and automated unit tests can be applied to ensure the correctness of the software.
Rating	PA
GP 2.1.1 Develop Work Products in an Iterative and Incremental Way	
Current Application	Software is developed in iterations. The games are divided into functional cohesive parts and each part is developed iteratively.
Strengths	Iterative development
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.1.2 Communicate Effectively	
Current Application	Team shares the same room and daily stand-up meetings (or as the team says grab-a-coffee) meetings are held to communicate and discuss daily activities and problems.
Strengths	Shared room Stand-up meetings
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
GP 2.2.1 Balance the Predictive Work and Adaptive Work	
Current Application	Work is balanced through Kanban board. Team commits to limiting WIP. Team plans cycles by choosing the work items that will be developed in that cycle.
Strengths	Kanban Methodology is used effectively to limit WIP, Cycle planning
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.2 Employ Minimally Sufficient Ceremonies and Specify Criteria for Documentation	
Current Application	The team has a policy that they're calling just-make-it-happen. The policy states that no ceremony should be held unless it's necessary. Also, team members state that they're applying RUP rules to the documents that is: "If document does not increase your understanding of the product then it's unnecessary".
Strengths	Ceremony is internally minimized throughout the team Document writing criteria is in place
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
GP 3.1.1 Incorporate Agile Engineering Methods/Practices to the Aspect Practices	
Current Application	Prototyping is used.
Strengths	Prototyping
Weaknesses	Correction of the solution is not verified
Improvement Suggestions	Pair programming and coding standards can be applied to increase quality of the code and TDD and unit tests can be used to ensure the correctness.
Rating	PA
GP 3.1.2 Integrate Tools to Aspects to Improve the Productivity	
Current Application	MS Visual Studio is used as an integrated development environment and GIT is used for configuration management.
Strengths	Tools are integrated for improved productivity
Weaknesses	N/A
Improvement Suggestions	N/A

Rating	FA
GP 3.2.1 Support Collaborative Work and Shared Responsibility	
Current Application	Team collaborates as a whole and it's self-organizing. Team members are sharing the responsibility of the game however they're organized according to their interests and expertize such as 3D design and voice engineering.
Strengths	Self-organizing teams
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.2 Adopt Agile Leadership Styles	
Current Application	Game owner leads the team with agile leadership styles and focuses on guiding people to do their works. Mistakes and problems are discussed without blaming and each mistake is embraced as a lesson learned.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

GP 3.2.3 Encourage People in the Organization to Participate in Learning, Teaching and Improvement	
Current Application	Mural is used for team members pointing the new technology that they should learn and apply. Also, teams constantly attend to educational events and conferences on the subjects such as UX Design and Agile Methodologies.
Strengths	Strengths are captured in Current Application
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

GP 3.2.4 Collect Measures to Support Learning and Improvement	
Current Application	No measures are kept for the construction aspect activities.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application
Improvement Suggestions	Code quality metrics such as defect density and check-in and check-out numbers can be collected and analyzed.
Rating	NA

Transition Aspect

T.AP1: Create and Manage the Development Workspace	
Current Application	There's only single environment for development and test. Code is under configuration control and changes are stated with comments when the code is committed.
Strengths	N/A
Weaknesses	There isn't an explicit testing approach. Changes to development items are not linked to other related artifacts.
Improvement Suggestions	A testing approach should be developed or adopted to ensure quality
Rating	PA
T.AP2: Integrate the Code	
Current Application	Code is integrated through check-in and check-out mechanisms.
Strengths	Strengths are captured in Current Application
Weaknesses	Integration frequency is not high enough
Improvement Suggestions	Frequency of integration can be increased. Automated integration and test mechanisms can be applied.
Rating	PA
T.AP3: Deploy the Solution	

Current Application	Deployment is performed however it's nor automatic nor continuous.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application
Improvement Suggestions	N/A
Rating	NA
T.AP4: Test the Integrated Solution	
Current Application	There's no explicit testing mechanism to test integrated solution. Tests are done manually via graphical user interface.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application
Improvement Suggestions	A testing approach can be developed and adopted to ensure correctness
Rating	NA
T.AP5: Make the Progress Visible	
Current Application	Transition process is visible to team members.
Strengths	Visible Transition Process

Weaknesses	Collaboration is not fully obtained.
Improvement Suggestions	Team members in testing and development could collaborate through the transition.
Rating	LA
T.AP6: Create the Supporting Documentation	
Current Application	Tutorials are prepared at the transition stage.
Strengths	Strengths are captured in Current Application
Weaknesses	Lack of maintenance documentation
Improvement Suggestions	Other documents for maintenance of the software can be produced.
Rating	PA
GP 2.1.1 Develop Work Products in an Iterative and Incremental Way	
Current Application	Transition activities are performed iteratively and incrementally. But, the iteration length is inconsistent.
Strengths	Strengths are captured in Current Application
Weaknesses	Weaknesses are captured in Current Application
Improvement Suggestions	Iteration length for the transition activities should be established.

Rating	LA
GP 2.1.2 Communicate Effectively	
Current Application	Team shares the same room and daily stand-up meetings (or as the team says grab-a-coffee) meetings are held to communicate and discuss daily activities and problems.
Strengths	Shared room, Stand-up meetings
Weaknesses	Collaboration is not fully obtained within the team
Improvement Suggestions	Team members in testing and development could collaborate through the transition.
Rating	LA
GP 2.2.1 Balance the Predictive Work and Adaptive Work	
Current Application	Limiting WIP principle seems to be not implemented to the transition aspect activities. The integration iteration lengths are inconsistent some are 4 weeks long some are a week long.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application
Improvement Suggestions	To balance the work iteration lengths should be established.
Rating	PA
GP 2.2.2 Employ Minimally Sufficient Ceremonies and Specify Criteria for	

Documentation	
Current Application	The team has a policy that they're calling just-make-it-happen. The policy states that no ceremony should be held unless it's necessary. Only documents created in this aspect are tutorials and integration comments.
Strengths	Strengths are captured in Current Application
Weaknesses	Lack of maintenance documentation and related criteria
Improvement Suggestions	Criteria for maintenance documents should be established to ease the maintenance.
Rating	LA
GP 3.1.1 Incorporate Agile Engineering Methods/Practices to the Aspect Practices	
Current Application	There aren't any Agile Practices applied for this aspect
Strengths	N/A
Weaknesses	N/A
Improvement Suggestions	Continuous integration, and Integration and acceptance tests should be performed.
Rating	NA
GP 3.1.2 Integrate Tools to Aspects to Improve the Productivity	
Current Application	GIT tool is used for deployment.

Strengths	Tool usage
Weaknesses	Deployment is not automated
Improvement Suggestions	Automatic deployment settings could be used.
Rating	PA
GP 3.2.1 Support Collaborative Work and Shared Responsibility	
Current Application	Team collaborates as a whole and it's self-organizing.
Strengths	Self-organizing teams
Weaknesses	Not all team members share the responsibility for deployment and integration. Only members who integrated the code are regarded responsible.
Improvement Suggestions	Responsibility of the deployment can be shared between members with consistent integration iterations that enables everyone to integrate their code.
Rating	PA
GP 3.2.2 Adopt Agile Leadership Styles	
Current Application	Game owner leads the team with agile leadership styles and focuses on guiding people to do their works. Mistakes and problems are discussed without blaming and each mistake is embraced as a lesson learned.
Strengths	Strengths are captured in Current Application

Weaknesses	N/A
Improvement Suggestions	Game leader should enable team members to make continuous deployments.
Rating	LA
GP 3.2.3 Encourage People in the Organization to Participate in Learning, Teaching and Improvement	
Current Application	Mural is used for team members pointing the new technology that they should learn and apply. Also, teams constantly attend to educational events and conferences on the subjects such as UX Design and Agile Methodologies.
Strengths	Strengths are captured in Current Application
Weaknesses	No learning objectives are present about continuous integration
Improvement Suggestions	Learning objectives on continuous integrations and integration and acceptance testing could be set.
Rating	LA
GP 3.2.4 Collect Measures to Support Learning and Improvement	
Current Application	No measures are collected during transition phase.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application

Improvement Suggestions	Defect analyzes and integration frequency could be collected and analyzed.
Rating	NA

Management Aspect

M.AP1: Initiate the Project	
Current Application	The feasibility study for the games are conducted based on the trend analysis. The popular game genres are examined and brainstorming is done to create ideas that conforms to the results of the trend analysis. Then ideas are recorded (as Game Design Documents) to the Mural tool and ranked based on voting.
Strengths	Trend Analysis Voting
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
M.AP2: Form the Team	
Current Application	Teams are formed based on the expertize areas. Teams include storyboard designers, graphic designers, software developers and sound artists.
Strengths	Strengths are captured in Current Application.
Weaknesses	N/A

Improvement Suggestions	N/A
Rating	FA
M.AP3: Align with Agile Values and Principles	
Current Application	Team and upper management are aligned with the Agile values and principles. Management embraces Agile and supports the practices.
Strengths	Strengths are captured in Current Application.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
M.AP4: Establish the Physical Work Space	
Current Application	Physical workspace has separate parts where team members can work together or work in isolation. Offices includes whiteboards, games and other elements to facilitate collaboration and thinking aloud.
Strengths	Strengths are captured in Current Application.
Weaknesses	N/A
Improvement Suggestions	N/A

Rating	FA
M.AP5: Plan the Progress	
Current Application	Release and sprint plans are prepared. Estimates are done based on the detailed screen since the time to develop similar screens are known. Daily stand-up meetings are held to monitor and improve the progress. Retrospective meetings are held to review the projects and discuss the lessons learned.
Strengths	Release and sprint plans, Estimations based on historical experiences
Weaknesses	Plans are not updated regularly through the changing conditions of the project.
Improvement Suggestions	Plans could be updated regularly with the changing requirements and conditions. Project Manager and Teams should steer the plans together.
Rating	LA
M.AP6: Estimate the Work Items	
Current Application	Estimates are done based on the detailed screen since the time to develop similar screens are known. There's a directory to keep actual past development efforts and times.
Strengths	Strengths are captured in Current Application.
Weaknesses	The actual values entered by the team members includes noise and causes deviations in estimates.
Improvement	A strategy for entering actual effort data could be

Suggestions	generated to reduce the noise.
Rating	LA
M.AP7: Monitor the Progress	
Current Application	Game Leaders monitor the progress of the team based on the screens completed and effort spent on them. The leader discusses these parameters regularly with the upper management.
Strengths	Strengths are captured in Current Application.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
M.AP8: Manage and Mitigate the Risks	
Current Application	Risk mitigation is not done.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application.
Improvement Suggestions	A risk tracking, analysis and mitigation strategy can be employed to manage and mitigate the risks.
Rating	NA

GP 2.1.1 Develop Work Products in an Iterative and Incremental Way	
Current Application	The plans, progress reports and project tracking activities are performed in multiple iterations.
Strengths	Strengths are captured in Current Application.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.1.2 Communicate Effectively	
Current Application	Daily stand-up meetings (or as the team says grab-a-coffee) meetings are held between Game Leader and team members to communicate and discuss daily activities and problems. Also, weekly progress meetings and retrospective meetings are held between Game Leaders and upper management.
Strengths	Shared room, Stand-up meetings
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 2.2.1 Balance the Predictive Work and Adaptive Work	

Current Application	Plans and estimations are not updated continuously during the course of the project.
Strengths	N/A
Weaknesses	Weaknesses are captured in Current Application.
Improvement Suggestions	Plans could be updated regularly with the changing requirements and conditions. Project Manager and Teams should steer the plans together.
Rating	NA
GP 2.2.2 Employ Minimally Sufficient Ceremonies and Specify Criteria for Documentation	
Current Application	Retrospective meetings enable non-value added activities to be eliminated from the process. Dashboard prints and automated reports are used as documentation to monitor and track the project progress.
Strengths	Strengths are captured in Current Application.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.1.1 Incorporate Agile Engineering Methods/Practices to the Aspect Practices	
Current Application	Estimations are done with collaboration between Game

	Leader and Team Members.
Strengths	Collaborative Estimation Making
Weaknesses	Plans and estimations are not updated continuously during the course of the project.
Improvement Suggestions	Plans and estimations should be updated continuously during the course of the project.
Rating	PA
GP 3.1.2 Integrate Tools to Aspects to Improve the Productivity	
Current Application	Mural and plan.io are used as management tools.
Strengths	N/A
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.1 Support Collaborative Work and Shared Responsibility	
Current Application	Estimation and planning are collaborative activities. Team members make their own estimates for the jobs they've assigned.
Strengths	Collaborative Estimation and Planning

Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.2 Adopt Agile Leadership Styles	
Current Application	Game owner leads the team with agile leadership styles and focuses on guiding people to do their works. Mistakes and problems are discussed without blaming and each mistake is embraced as a lesson learned.
Strengths	Strengths are captured in Current Application.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA
GP 3.2.3 Encourage People in the Organization to Participate in Learning, Teaching and Improvement	
Current Application	Game leaders share their knowledge and experiences through retrospective meetings to learn from each other.
Strengths	Strengths are captured in Current Application.
Weaknesses	Learning objectives are not defined.

Improvement Suggestions	Management aspect based learning objectives can be set.
Rating	LA
GP 3.2.4 Collect Measures to Support Learning and Improvement	
Current Application	Management aspect activities are tracked and monitored through measures.
Strengths	Strengths are captured in Current Application.
Weaknesses	N/A
Improvement Suggestions	N/A
Rating	FA

TEZ FOTOKOPİ İZİN FORMU

ENSTİTÜ

Fen Bilimleri Enstitüsü ☐

Sosyal Bilimler Enstitüsü ☐

Uygulamalı Matematik Enstitüsü ☐

Enformatik Enstitüsü ☐

Deniz Bilimleri Enstitüsü ☐

YAZARIN

Soyadı : .....
Adı : .....
Bölümü : .....

TEZİN ADI (İngilizce) : .....
.....
.....
.....
.....

TEZİN TÜRÜ : Yüksek Lisans ☐ Doktora ☐

1. Tezimin tamamı dünya çapında erişime açılsın ve kaynak gösterilmek şartıyla tezimin bir kısmı veya tamamının fotokopisi alınsın. ☐
2. Tezimin tamamı yalnızca Orta Doğu Teknik Üniversitesi kullancılarının erişimine açılsın. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐
3. Tezim bir (1) yıl süreyle erişime kapalı olsun. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐

Yazarın imzası ..... Tarih .....